The MANGA Data Analysis Pipeline: a prototype

L. Coccato & MANGA team

February 17, 2014

Contents

Chapter 1

Introduction

1.1 Data Analysis Pipeline (DAP): overview

The scope of the Data Analysis Pipeline is to analyse the output of the Data Reduction Pipeline and deliver science products. They are classed into two groups, the **High level data products** and the **Model dependent data products**.

High level data products are:

- Stellar kinematics (velocity, velocity dispersion, h3, and h4 Gauss-Hermite moments).
- Emission line kinematics (velocity, and velocity dispersion).
- Equivalent widths and fluxes of emission lines.
- Absorption line strengths.
- Reddening.
- Radial gradients of measured quantities.
- Rotation curves and kinemetric parameters.
- Stellar weights from full spectral fitting.

Model dependent data products are:

- Stellar population parameters age, metallicity, chemical element ratios, and IMF, and their radial gradients (from absorption features).
- Stellar population parameters age, metallicity, chemical element ratios, and IMF, and their radial gradients (from full spectral fitting).
- Extinction corrected star formation rates and hystories.
- Gas metallicities, BPT diagrams.
- Mass to light ratio (from stellar population).
- Mass to light ratio, dynamical mass (from dynamical modeling).
- Angular momentum.

1.1.1 Installation and requirements

T.B.D.

Table 1.1: Summary of blocks, interfaces and main modules distribution.

BLOCK	Interface modules	Main modules	Utility modules
Block 1	mdap_read_datacube	mdap_calculate_spectrum_sn	
Block 2	mdap_spatial_binning	mdap_voronoi_2d_binning	mdap_calibrate_sn
Block 3	mdap_log_rebin[1]	mdap_do_log_rebin	mdap_convol_sigma
Block 4	mdap_spectral_fitting mdap_create_starting_guesses	mdap_calculate_spectrum_sn mdap_bvls(_external) mdap_dust_calzetti mdap_get_losvd mdap_gandalf_wrapf[2] mdap_gandalf[2] mdap_ppxf[2] `mpfit package`	mdap_range.pro mdap_stc.pro mdap_sgn.pro mdap_round_str.pro mdap_interpolate_2dmaps mdap_ppxf_convol_fft
Block 5	mdap_measure_indices	mdap_read_indices_definitions[3] mdap_do_measure_index	mdap_round_str.pro mdap_convol_sigma
Block 6	mdap_spatial_radial_binning mdap_spectral_fitting mdap_measure_indices	mdap_calculate_spectrum_sn mdap_do_measure_index mdap_dust_calzetti `mpfit package` mdap_sgandalf mdap_gandalf_wrapf[2] mdap_gandalf[2] mdap_ppxf[2] mdap_bvls(_external)	mdap_convol_sigma mdap_get_losvd mdap_range.pro mdap_stc.pro mdap_sgn.pro mdap_interpolate_2dmaps mdap_read_indices_definitions[3] mdap_ppxf_convol_fft
Block 7			
Block 8			
Block 9			

Notes: [1] Need auxiliary files: stellar templates. [2] Need auxiliary file: Emission lines definitions. [3] Need auxiliary file: Absorption line index definitions.

1.2 DAP workflow description

The DAP is divided into parts and blocks, the first part (blocks 1-6) will deliver the High Level Data products (see Figure 1.1), the second part (blocks 7-9) will deliver the Model dependent data products (see Figure 1.2).

Each block is responsable for a set of operations, such as reading data, fitting the input spectra. The procedure that executes an operation is called "main module". The main modules comunicates between them and between the various blocks via other procedures, which are called "interface module". The main and interfaces modules can call other modules, which are called "unitily modules". The interface module is therefore responsable to get input files from a previous interface, convert them in a readable format readable for the main module, collect the outputs of the main module and convert them in a form readable for the next interface. In this way, it is relatively easy to change the software responsible for a specific task (i.e. replacing the main module responsible for the spectral fitting) by chaning its interface. Table 1.2 list a summary of the blocks, interfaces and main modules distribution.

The input datacube is read in block 1 and information (vectors with galaxy spectra, errors, wavelenght, and spatial information) are passed to the block 2 for spatial binning.

Three spatial binnings are foreseen, depending on the scientific requirements. The current DAP version uses the Voronoi binning scheme, (as implemented in IDL by Cappellari & Copin 2003), as main module for the spatial binning task.

Binned spectra are passed to block 3 for logarithmic resampling of the galaxy spectra (and error) and the stellar templates. Stellar templates are also broadened to match the instrumental set up. For this, the instrumental $LSF(\lambda)$ is required as input. Three sets of log-sampled galaxy spectra are produced, one set for each spatial binning.

The log-sampled spectra are then passed to block 4 for spectroscopic measurements. Three fits are performed in block 4, one for each spatial sampling defined in block 2. Before each fit, the log-sampled Galaxy spectra are corrected for Milky Way extinction(input parameter). Results from the first execution will be used to constrain the fit of the second fit, and soforth. The current version uese: i) the pPPXF.pro and gandalf.pro (Cappellari & Emsellem 2004; Sarzi et al. 2006); and ii) Calzetti et al. (2000) formulas for reddening correction as main modules in block 4 (See Section 2.4.2 for further details). Fitting procedures have been modified to allow the use of external fortran routine, instrumental velocity dispersion variable with wavelength, and fitting parameter boundaries from user input.

The output of block 4 are the kinematic parameters of stars and gas, emission lines fluxes and equivalent width, reddening, the weights of the stellar templates used in the fit (for stellar population measurements), and rest-framed galaxy spectra.

Input galaxy spectra (with best-fit emission lines removed) will be passed to block 5 for measurement of the line strenght. The current design foresees that absorption line strength will be measured only onto spectra associated to the first spatial binning (i.e. those with the higher S/N. The current version uses the absoprtion line indices as defined by Worthey et al. (1994).

Rest-framed spectra, Kinematic measurements, emission line fluxes, absorption and emission line equivalent widths are then passed to block 6 for the extraction of the radial profiles of the measured quantities, and kinemetric analysis.

1.2.1 DAP inputs and outputs

The Data Analysis Pipeline consists in a IDL procedure, manga_dap.pro, and a set of "interfaces" modules, "main" modules, and "utilities", which are described in this document.

To run the pipeline, the following files are needed.

- total_filelist.dat. This is the file that specifies the galaxies to analyse and their properties. The file must contain 5 columns. The first column indicates the names of the N datacubes (stored in .fits file, see Section 2.1.1), the second column provides an estimate of the galaxy redshift (in km/sec), the third column provides an estimate of the stellar velocity dispersion (in km/sec), the 4th column provides the mean galaxy ellipticity, and the 5th column provides the mean galaxy position angle.

- all the N files (datacubes or rss) listed in the file total_filelist.dat.

- a set of stellar templates, in fits file format.

- A file containing the definitions of the emission lines to include in the fit (see Section **??**).

- A file containing the definitions of the absorption line indices to measure (see Section 3.3.2).
- A configuration file, that contains all the parameters needed in the analysis.

The DAP is executed by the following IDL command line

$$\texttt{IDL} > \texttt{manga\_dap}, \texttt{i}, \texttt{configuration\_file}$$

where `i` is the index number of the i-th entry in the total_filelist.dat, column 1 to analyse, $i = 0, N-1$. configuration_file is a string specifying the name of the configuration file. For a full description of the configuration file, see Section 1.3.

As output, the DAP returns a multilayer fits file with all the measured quantities ((`<datacube_name>_high_level.fits`), an idl session with all the session variables stored (`<datacube_name>_mdap_session.idl`). and a log file (¡datacube_name¿_mdap.log).

The content of the `<datacube_name>_high_level.fits` output file is described in Table 1.2.

Table 1.2: Extension description of the DAP output fits file.

Ext	Name	Description
0	signal	Mean signal per pixel, produced by mdap_read_datacube.pro (Section 2.1.1).
1	noise	Mean noise per pixel, produced by mdap_read_datacube.pro (Section 2.1.1).
2	binning map 1	Location and geometry of the spatial bins of the first binning scheme.
3	binning 1 data	Measurements performed on the first binning scheme (absorption line indices).
4	binning map 2	Location and geometry of the spatial bins of the second binning scheme.
5	binning 2 data	Measurements performed on the second binning scheme (stellar kinematics).
6	binning map 3	Location and geometry of the spatial bins of the third binning scheme.
7	binning 3 data	Measurements performed on the third binning scheme (gas kinematics and emission line properties).
8	binning map radial	Location and geometry of the radial binning scheme.
9	binning radial data	Measurements performed on the radial binning scheme (absorption line indices and stellar velocity dispersion)

Extension 3: outputs related to the first binning scheme

The quantities measured and stored using the first binning scheme are the following:

- Column 1. X. The X coordinates (in arcsec) of the centers of the spatial bins. The

center of the field of view has coordinates (0,0).

- Column 2. Y. The Y coordinates (in arcsec) of the centers of the spatial bins. The center of the field of view has coordinates (0,0).
- Column 3. AREA_BIN. Area in arcsec^2 of the spatial bin.
- Column 4. STON. Estimate of the S/N of the spectrum in the spatial bin. The signal is defined as the median of the best fit model, the noise as the robust_sigma of the residuals (observed spectrum - best fit).
- Column 5. NELEMENTS. Number of spectra coadded in the spatial bin.
- Columns 6-end. Equivalent width of the absorption line indices and their errors (Units Å or magnitudes, depending on the index definition. The measured indices (and their names) are defined in a user-provided file. The name of this file is specified in the configuration file (see Section 1.3).

Extension 5: outputs related to the second binning scheme

The quantities measured and stored using the second binning scheme are the following:

- Column 1. X. The X coordinates (in arcsec) of the centers of the spatial bins. The center of the field of view has coordinates (0,0).
- Column 2. Y. The Y coordinates (in arcsec) of the centers of the spatial bins. The center of the field of view has coordinates (0,0).
- Column 3. AREA_BIN. Area in arcsec^2 of the spatial bin.
- Column 4. STON. Estimate of the S/N of the spectrum in the spatial bin. The signal is defined as the median of the best fit model, the noise as the robust_sigma of the residuals (observed spectrum - best fit).
- Column 5. NELEMENTS. Number of spectra coadded in the spatial bin.
- Columns 6-7. VEL and VEL_ERR. Measured stellar velocity and its error in km/sec.
- Columns 8-9. DISP and DISP_ERR. Measured stellar velocity dispersion and its error in km/sec.
- Columns 10-11. H3 and H3_ERR. Measured Gauss-Hermite moment of the stellar velocity distribution h3 and its error (Warning: the maximum range allowed in the fitting procedure is : $-0.4 < \mathrm{H3} < 0.4$).
- Columns 12-13. H4 and H4_ERR. Measured Gauss-Hermite moment of the stellar velocity distribution h4 and its error (Warning: the maximum range allowed in the fitting procedure is : $-0.4 < \mathrm{H4} < 0.4$).
- Column 14. CHI2. Chi-squared from the fit.

Extension 7: outputs related to the third binning scheme

The quantities measured and stored using the third binning scheme are the following:

- Column 1. X. The X coordinates (in arcsec) of the centers of the spatial bins. The center of the field of view has coordinates (0,0).
- Column 2. Y. The Y coordinates (in arcsec) of the centers of the spatial bins. The center of the field of view has coordinates (0,0).
- Column 3. AREA_BIN. Area in arcsec^2 of the spatial bin.
- Column 4. STON. Estimate of the S/N of the spectrum in the spatial bin. The signal is defined as the median of the best fit model, the noise as the robust_sigma of the residuals (observed spectrum - best fit).
- Column 5. NELEMENTS. Number of spectra coadded in the spatial bin.
- Columns 6-7. VEL and VEL_ERR. Flux-weighted mean velocity of the emission lines and its error in km/sec.
- Columns 8-9. DISP and DISP_ERR. Flux-weighted mean velocity dispersion of the emission lines and its error in km/sec.
- Columns 10-11. E(B-V) color excess for the stellar component and its error
- Columns 12-13. E(B-V) color excess for the ionized gas stellar component and its error
- Column 14. CHI2. Chi-squared from the fit.
- Columns 15-end. Flux, flux error, Equivalent widths, and equivalent width errors of the emission lines. The measured emission lines (and their names) are defined in a user-provided file. The name of this file is specified in the configuration file (see Section 1.3).

Extension 9: outputs related to the radial binning scheme

The quantities measured and stored using the radial binning scheme are the following:

- Column 1. AMAJ. Lenght of the semi-major axis (in arcsecs) describing the elliptical bin. AMAJ=0 is the center of the field of view.
- Column 2. AMAJ_LO. Lower limit boundary (in arcsec) of elliptical bin.
- Column 3. AMAJ_UP. Upper limit boundary (in arcsec) of elliptical bin.
- Column 4. STON. Estimate of the S/N of the spectrum in the spatial bin. The signal is defined as the median of the best fit model, the noise as the robust_sigma of the residuals (observed spectrum - best fit).
- Columns 5-6. DISP and DISP_ERR. Measured stellar velocity dispersion and its error in km/sec.

- Column 7. CHI2. Chi-squared from the fit.
- Columns 8-end. Equivalent width of the absorption line indices and their errors (Units Å or magnitudes, depending on the index definition. The measured indices (and their names) are defined in a user-provided file. The name of this file is specified in the configuration file (see Section 1.3).

1.3 The configuration file

The configuration file defines variables and parameters used in the DAP. No empty lines should be present, commented lines are marked with '#'. The content of the configuration file is the following:

- total_filelist. A string indicating the full path to the file listing the galaxies to analyze and their physical parameters (See Section 1.2.1).
- datacube_root_dir. Path indicating the location of the data to analyse. Files in datacube format must be stored in <datacube_root_dir>\datacubes; files in RSS format must be stored in <datacube_root_dir>\rss.
- output_root_dir. Path indicating where the results of the analysis should be stored. The directories <output_root_dir>\resuts_datacubes, and <output_root_dir>\resuts_rss must exist.
- w_range_for_sn_computation. Two elements array that specifies the wavelength range where to compute the signal to noise ratio. This array is passed to the main module mdap_read_datacube.pro, via the optional input `lrange` (See Section 2.1.1). The suggested vaue for MANGA is to adopt the r-gunn FWHM bandpasse, centered at the effective wavelength (i.e. $5560.00 < \lambda < 6942.00$ Å Fukugita et al. 1998). Leave it undefined to use the entire spectral range.
- w_range_for_sn_computation_for_gas. Two elements array that specifies the wavelength range where to compute the signal to noise ratio for emission line science. This range will be redshifted accordingly to the galaxy systemic velocity (from the starting guesses). If not specified, the default option will be adopted, i.e. use the same wavelenght range (with no redshift correction) for continuum and emission line science. One possible suggestion is to use the $6530 < \lambda < 6600$ wavelength range, which embraces Hα and NII emission lines.
- trim_wav_range_spatial_binning_1. Two elements vector specifying the wavelength region to analyse in the first spatial binning (units: Å). Binned galaxy spectra (of the first spatial binning) will be trimmed accordingly, templates will be trimmed over a sligtly larger wavelenth range. Its value is passed to the main module mdap_log_rebin, during the rebinning and trimming of the spectra in the first spatial binning scheme through the optional input wave_range (See Section 2.2.1). If not defined, the entire available wavelength range will be used.
- trim_wav_range_spatial_binning_2. As above, but for the second spatial binning scheme (3 spatial binning schemes are foreseen in the DAP, see Section 2.2, plus a radial binning scheme 2.6).
- trim_wav_range_spatial_binning_3. As above, but for the third spatial binning scheme (3 spatial binning schemes are foreseen in the DAP, see Section 2.2, plus a radial binning scheme 2.6).
- trim_wav_range_radial_binning. As above, but for the radial spatial binning scheme (3 spatial binning schemes are foreseen in the DAP, see Section 2.2, plus a radial binning scheme 2.6).

- velscale. Float value indicating the velocity scale to adopt for the logarithmic rebinned spectra (units km/sec/pixel). This value is passed to mdap_log_rebin via the keyword input_velscale (see Section 2.3.1), and to mdap_spectral_fitting via the velscale input variable (see Section 2.4.2). The default is to use the one automatically defined by the input galaxy specta. Suggested value: 30 km/sec/pixel.

- stellar_library_spatial_binning_1. String specifying the path. Warning: files are identified with the IDL function `file_search(stellar_library_spatial_binning_1)`. Therefore, be sure that this is enough to identify all and only the files that are needed. Warning: stellar_library_spatial_binning_1 will be used also for the radial binning scheme. Files in the library must be fits files covering a wavelenght range preferentially larger than the MANGA ($3000 < \lambda < 10000$). They must have an uniform angstrom/pixel sampling. The following header keywords need to be present: CRVAL1 (value at reference pixel), CRPIX1 (reference pixel), and CDELT1 (dispersion in angstrom/pixel). stellar_library_spatial_binning_1 is required, and it will be passed to the interface mdap_log_rebin and via the library input variable.

- stellar_library_spatial_binning_2 Same as stellar_library_spatial_binning_1, but for stars to be used with the spectra of the secon spatial binning.

- stellar_library_spatial_binning_3. Same as stellar_library_spatial_binning_1, but for stars to be used with the spectra of the third spatial binning.

- sn1_rss. Float indicating the target signal-to-noise to adopt in the Voronoi binning scheme for RSS format data in the first spatial binning. This is mandatory and it will be passed to the interface mdap_spatial_binning via the min_sn input variable (see Section 2.2.1). Suggested entry = 15.

- sn2_rss. Same of sn1_rss, but for RSS format data in the first spatial binning. Suggested entry = 10.

- sn3_rss. Same of sn1_rss, but for RSS format data in the third spatial binning. Suggested entry = 5.

- sn1_datacubes. Same of sn1_rss, but for datacube format data in the first spatial binning. Suggested entry = 40.

- sn2_datacubes. Same of sn1_rss, but for datacube format data in the second spatial binning. Suggested entry = 25.

- sn3_datacubes. Same of sn1_rss, but for datacube format data in the third spatial binning. Suggested entry = 15.

- sn_thr_tpl_rss = 2. Threshold value for the S/N per angstrom each spectrum of RSS data format needs to have to be included in the analysis. S/N is computed over the wavelength range defined by the w_range_for_sn_computation variable (see above). Specra whose S/N are lower than this value will be discarded.Default = 0. To avoid any S/N threshold, set this variable to a very negative value (i.e. -100). Suggested value = 2

- sn_thr_str_rss. Same as sn_thr_tpl_rss, but for RSS format data in the second spatial binning scheme. Suggested value = 2.

- sn_thr_ems_rss. Same as sn_thr_tpl_rss, but for RSS format data in the hird spatial binning scheme. Suggested value = 2.

- sn_thr_tpl_datacubes. Same as sn_thr_tpl_rss, but for datacube format data in the first spatial binning scheme. Suggested value = 2.

- sn_thr_str_datacubes. Same as sn_thr_tpl_rss, but for datacube format data in the second spatial binning scheme. Suggested value = 2.

- sn_thr_ems_datacubes. Same as sn_thr_tpl_rss, but for datacube format data in the third spatial binning scheme. Suggested value = 2.

- sn_calibration_rss. Calibration coefficients for RSS format files. Formula is specified in the mdap_calibrat_sn.pro function. If not defined no calibration will be used (seggested). sn_calibration_rss is passed to mdap_spatial_binnin.pro and mdap_voronoi_2d_binning.pro via the sn_calibration keyword

- sn_calibration_datacubes. Calibration coefficients for datacube format files. Formula is specified in the mdap_calibrat_sn.pro function, and it is given by the following expression:

$$\mathrm{S/N_{REAL}} = \sum_{i=1,N} C_i \cdot \left(\frac{\mathrm{S/N_{ESTIMATED}}^{C_0}}{\sqrt{\mathrm{N_{spax}}}} \right)^{i-1}$$

  where C_i are the coefficients of sn_calibration_datacubes, and $\mathrm{N_{spax}}$ are the number of spaxels in that spatial bin.

  If sn_calibration_datacubes is not defined, the relation $\mathrm{S/N_{REAL}} = \mathrm{S/N_{ESTIMATED}}$ is used. Suggested values for the MaNGA test run are: [1.1, 0.743865, 1.10317, –0.0106751, $4.00892 \cdot 10^{-5}$].

  Warning: Coefficient calibrations were not computed for optimally weighted binned spectra (i.e. do not use if weight_for_sn=1.)

  sn_calibration_datacubes is passed to mdap_spatial_binnin.pro and mdap_voronoi_2d_binning.pro via the sn_calibration keyword (see Section 2.2.1).

- weight_for_sn. If set to 0, the voronoi binning scheme uses the Modified Lyoid algorithm, and spectra belonging to the same spatial bin are added together with no weighting. If set to 1 the voronoi binning scheme uses an optimal weighting procedure (Signals and Errors is weighted by S/N^2), and spectra belonging to the same spatial bin are added with weights given by S/N^2. This value is passed to the interface mdap_spatial_binning.pro and the module mdap_voronoi_2d_binning.pro via the keyword weight_for_sn (see Section 2.2.1). If not defined, the default value (0) is used (no weights are applied). Warning: if the weighting is used, the suggested S/N calibration (see below) is not valid.

- user_bin_map_spatial_binning_1. String specifying the fits file to be used to set the first spatial binning scheme. If provided, it will override the Voronoi binning scheme. Default: use the S/N to define the voronoi binning. Its value will be passed to the interface mdap_spatial_binning.pro via the keyword user_bin_map (see Section 2.2.1)

- user_bin_map_spatial_binnin_2. Same as before, but for the second spatial binning scheme.
- user_bin_map_spatial_binnin_3. Same as before, but for the third spatial binning scheme.
- emission_line_file_spatial_binnin_1. String specifying the name of the file which defines the emission lines to be fitted in the first spatial binning and the fit set-up parameters (see Sections 2.4.2 and 3.4). Its value will be passed to the interface mdap_spectral_fitting.pro via the keyword emission_line_file (see Section 2.4.2).
- emission_line_file_spatial_binnin_2. Same as before, but for the second spatial binning.
- emission_line_file_spatial_binnin_3. Same as before, but for the third spatial binning.
- emission_line_file_radial_binning. Same as before, but for the radial binning.
- absorption_line_indices. String specifying the name of the file which defines the absorption line indices to measure. It must be an ascii file with 9 columns:
  - Column 1. Integer. Unique ID number of the absorption line feature.
  - Column 2.String. Unique name of the absorption line feature. This will define the name of the field in sctructure of the DAP results (i.e. the name must begin with a letter, special characters like comas or dots are not allowed).
  - Columns 3-4. Float (units: Å) Lower and upper values of the index passpand.
  - Columns 5-6. Float (units: Å) Lower and upper values of the index blue pseudo-continuum.
  - Columns 7-8. Float (units: Å) Lower and upper values of the index red pseudo-continuum.
  - Column 9. String (accepted values are: ang or mag). Specifies the units (Å or magnitudes) of the output.

  absorption_line_indices must be defined, and it will be passed to the interface mdap_measure_indices.pro and the utility mdap_read_indices_definitions.pro via the absorption_line_indices input variable (see Sections **??** and **??**).

  Indices will be measured only if their blue and red pesudo-continua bandpasses are included in the considered wavelength range. If not, their values are set to NaN, and their errors to 99 in the final output file.
- save_intermediate_steps. If set to 1 at the end of each block an idl session with all the current variables is saved. Default: 0 (suggested)
- remove_null_templates. If set to 1, the stellar templates that have null weight after the fiting of the first spatial bin will be rejected from the other fits. It is meaningful only if the libraries used for the other spatial binnings are the same of that used for the first spatial binning. Default: 0 Suggested value = 1.
- external_library. String indicating the location of fortran or C executables that will replace some of the IDL modules for speed purposes. If undefined, IDL routines will be used. The current DAP structure foresees the following two values, and mdap_bvls.pro will be substituted by the mdap_bvls_external.pro routine:

1. external_library= <mangadap>/external/F90_32/ for 32 bit machines;
2. <mangadap>/external/F90_64/ for 64 bit machines.

Default: undefined (using internal routines). we suggest to define it and use external routines. The value of external_library will be passed to the interface mdap_spectral_fitting.pro via the keyword external_library (if defined it will be used for all the spatial binning)

- spectra_fittin_parameters_patial_binning_1. String array containing the additional inputs and keyword to be passed to the mdap_spectral_fitting.pro interface via the input keyword extra_inputs (see Section 2.4.2) to be used to analyse the spectra of the first spatial binning. Suggested values: `['MOMENTS = 4', 'DEGREE = -1', 'MDEGREE = 4']` Warning: variables will be defined via the IDL execute() function within the mdap_spectral_fitting.pro.

- spectra_fittin_parameters_patial_binning_2. Same of before, but for the second spatial binning. Suggested value: `['MOMENTS = 4', 'DEGREE = -1', 'MDEGREE = 4']`

- spectra_fittin_parameters_patial_binning_3. Same of before, but for the third spatial binning. Suggested value: `['MOMENTS=4', 'MDEGREE=4', 'DEGREE=-1', 'reddening = [0.01,0.01]', 'LAMBDA = exp(loglam_gal)']`

- spectra_fittin_parameters_patial_binning_readial. Same of before, but for the radial binning. Suggested value: `['MOMENTS=4', 'DEGREE=-1', 'mdegree=4', 'reddening = [0.01]', 'LAMBDA = exp(loglam_gal)']`

- instrumental_fwhm_file String indicating the name of the file with the wavelenght dependence of the instrumental FWHM. The file needs to be ascii with 4 columns.

  - Column 1 : wavelength in angstrom, where the instrumental FWHM has been measured
  - Column 2 : Resolving power.
  - Column 3 : Instrumental FWHM in angstrom.
  - Column 4 : Instrumental FWHM in km/sec.

  Warning: Column 2 will not be used by the DAP, but it needs to be present to make the procedure running.

  If instrumental_fwhm_file is undefined, a constant instrumental FWHM of 2.54 Å will be used.

1.3.1 Version control and tags. Obsolete

At the beginning of the execution, the DAP check whether the output .fits file and .idl sessions exist. If so, it reads the version(s) used to generate the existing files. If the current module versions are greater than those used to generate the output file, the analysis is performed again.

- **manga_dap_version**. The version of the DAP. If the module version is greater than that stored in the output files, the entire analysis is performed again (currently, blocks 1-5).
- **mdap_read_datacube_version**. The version of modules in block 1. If it is greater than that stored in the output file, blocs 1-5 are executed again.
- **mdap_spatial_binning_version**. The version of modules in block 2. If it is greater than that stored in the output file, blocs 2-5 are executed again. It must include version information about mdap_voronoi_binning.pro as well.
- **mdap_log_rebin_version**. The version of modules in block 3. If it is greater than that stored in the output file, blocs 3-5 are executed again.
- **mdap_spectral_fitting_version**. The version of modules in block 4. If it is greater than that stored in the output file, blocs 4-5 are executed again. It must include version information about mdap_sgandalf.pro as well.
- **mdap_measure_index_version**. The version of modules in block 5. If it is greater than that stored in the output file, block 5 is executed again.

Warning: the version control is enabled only if the keyword /check_version is set.

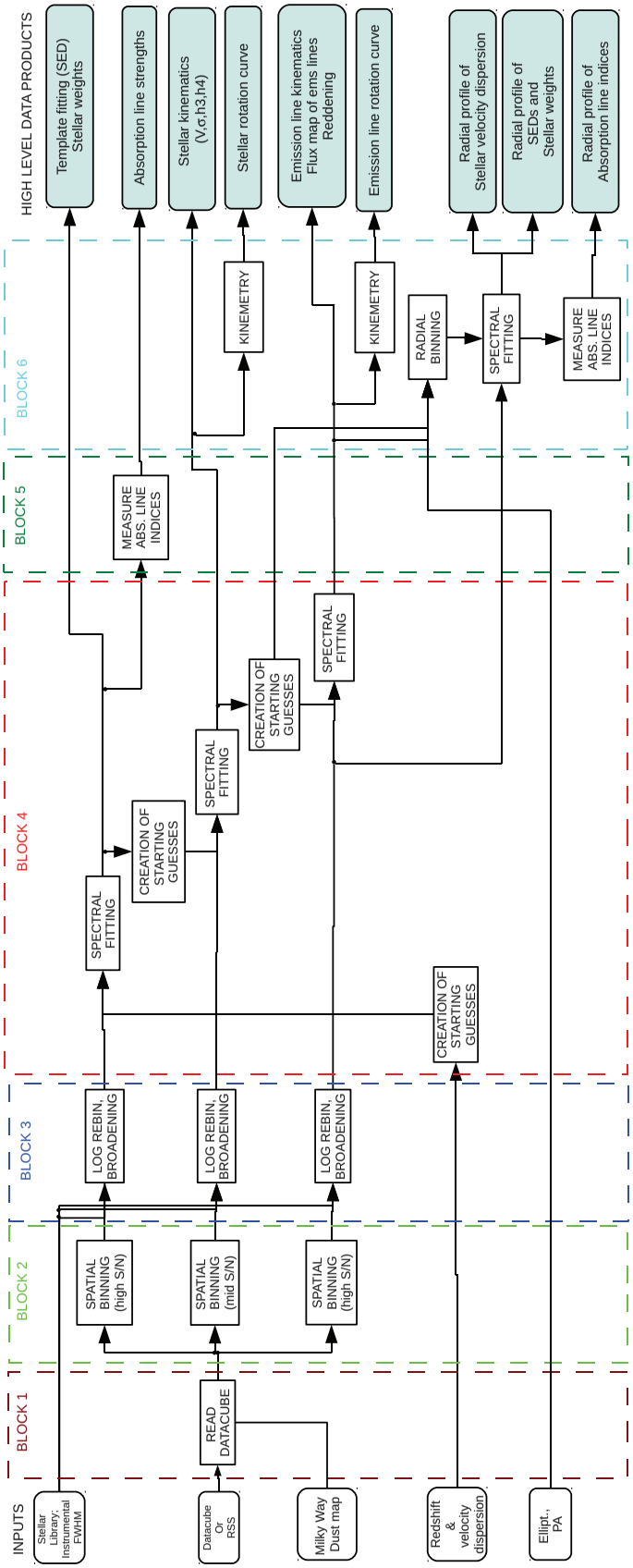

Figure 1.1: Workflow chart of the Data Analysis Pipeline (High Level Data products).

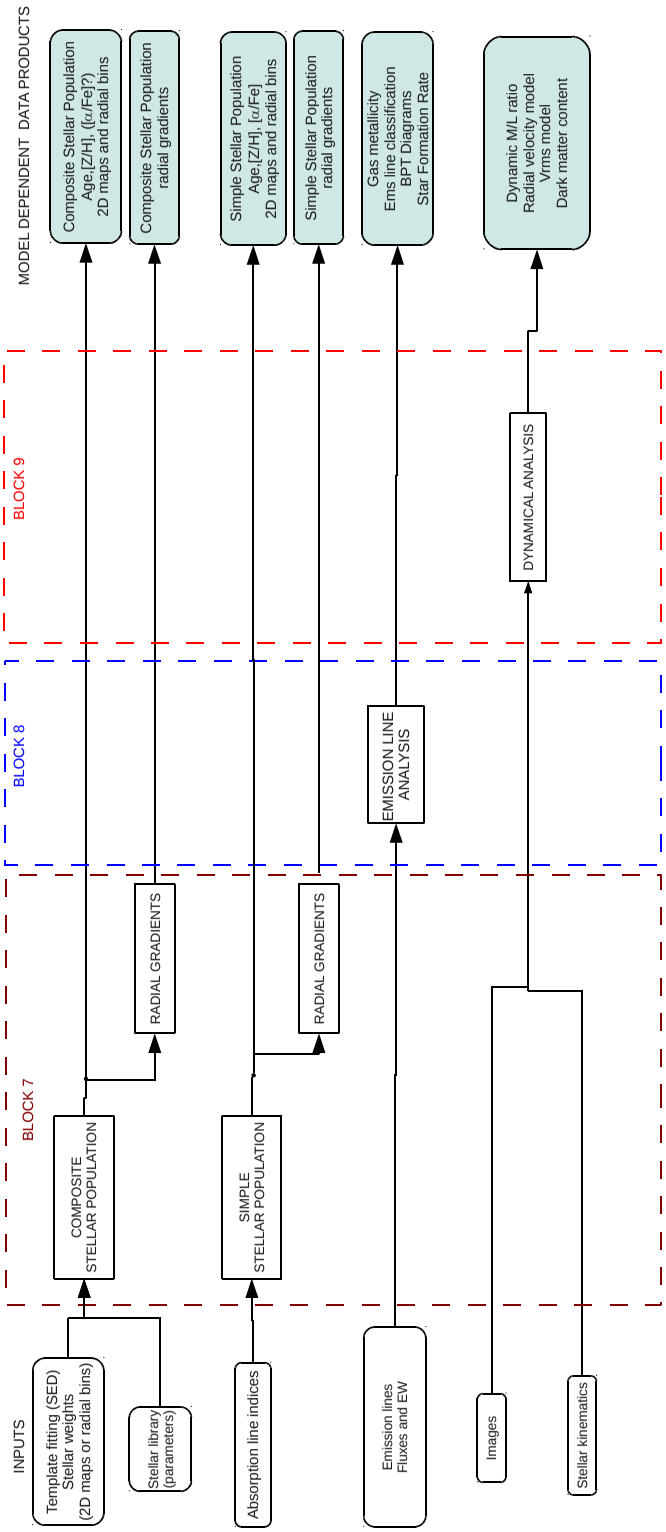

Figure 1.2: Workflow chart of the Data Analysis Pipeline (Model Dependent Data products).

Chapter 2

BLOCKs and "interface" modules

2.1 DAP Block 1: reading the input data

This block reads the input data (in the form of a datacube or RSS files) and extracts the all information needed for further processing, such as the galaxy spectra and errors, wavelength vector, 2D maps containing the coordinates, and galaxy signal and noise. The interface module in this block is: mdap_read_datacube.pro (see Section 2.1.1 for a list of inputs and outputs).

The main modules in the block, called by the interface, are mdap_calculate_spectrum_sn.pro (see Section 3.1 for a list of inputs and outputs).

The Block will provide the galaxy spectra (and errors), two-dimensional maps with coordinates (in arcseconds, the center of the field has coordinates 0,0), the two-dimensional maps of the signal and the noise (per angstrom), the wavelength vector (in Å) and related quantities (initial Å, reciprocal dispersion Å pixel^{-1}), and header information for 2-dimensional maps. Figure 2.1 illustrates the flowchart of block 1.

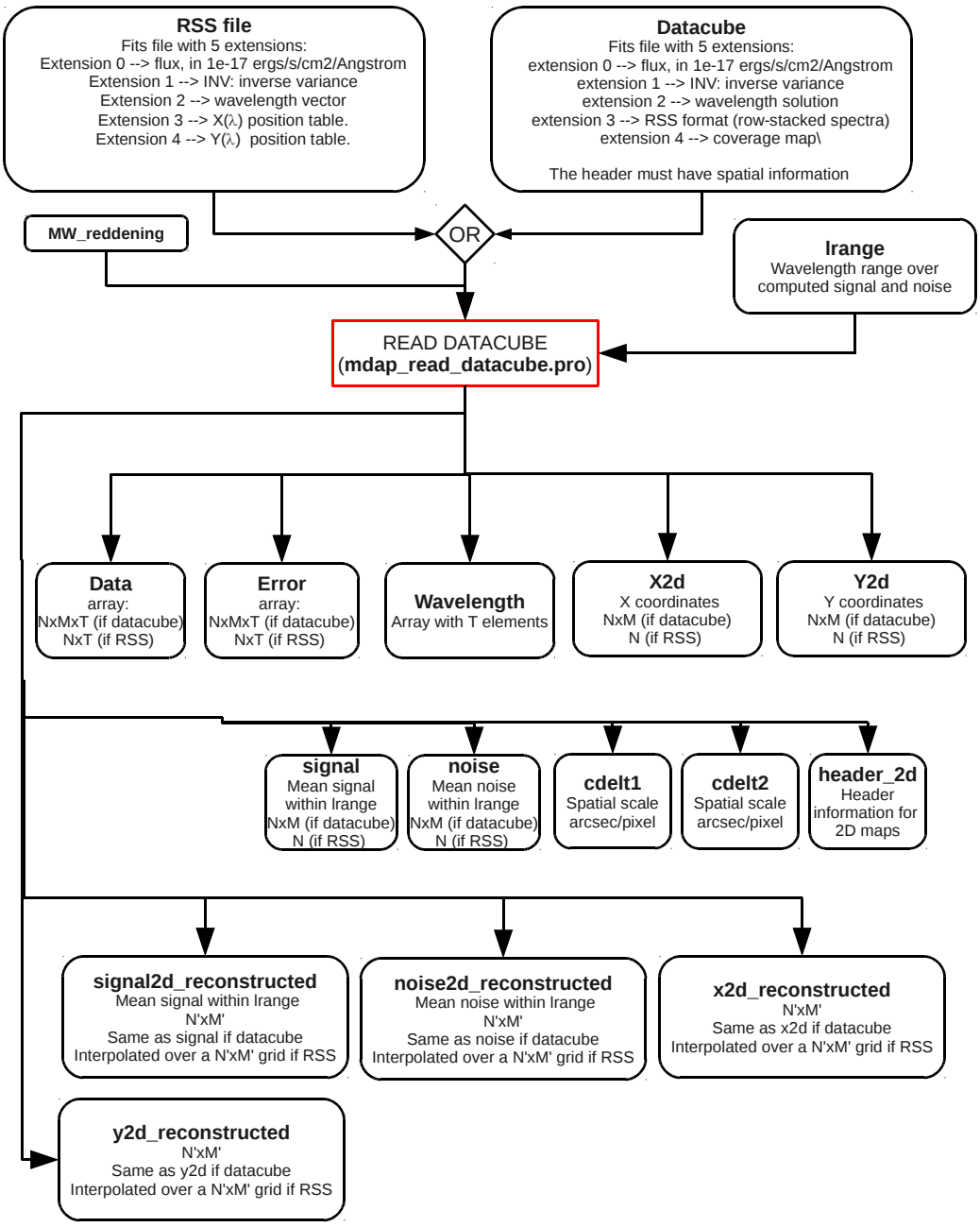

Figure 2.1: Workflow chart of the block 1 of the Data Analysis Pipeline. See Section 2.1.1 for a description of inputs and outputs.

2.1.1 `mdap_read_datacube.pro`

This interface module reads the input datacube or RSS files (which must be multi-layer fits file) and extract from it all the arrays and quantities, which are needed for the analysis. Table 2.1 list the inputs and outputs required for this module.

Table 2.1: Inputs and outputs parameters of mdap_read_datacube.pro

INPUTS	
datacube_name	string with the name of the fits file containing the input in datacube or RSS formats. For inputs in datacube format, the file must be a fits file with the following extensions: 1. flux in 1e-17 ergs/s/cm^{-2}/Å. This extension must be a 3D array, with the wavelength direction along the 3rd axis. 2. Inverse variance associated to the first extension 3. Wavelength solution (1D dbl array). Constant linear step (preferred)? Constant Log-step? Constant ln-step? 4. RSS format (row-stacked spectra) (NOT USED) 5. coverage map; (NOT USED) For inputs in RSS format, the file must be a fits file with the following extensions: 1. flux in 1e-17 ergs/s/cm^{-2}/Å. This extension must be a 3D array, with the wavelength direction along the 3rd axis. 2. Inverse variance associated to the first extension 3. Wavelength solution (1D dbl array). Constant linear step (preferred)? Constant Log-step? Constant ln-step? 4. X position table. Since position is a function of wavelength this is an array of the same size as the flux array. Coordinates should be in arcseconds relative to the center of the data cube (i.e., roughly the center of the galaxy). 5. Y position table.
OPTIONAL INPUTS	
lrange	[2 elem vector]. It indicates the wavelentgh range (in angstrom) where to extract the information for Signal and Noise. Default: use the entire spectra range
OPTIONAL KEYWORDS	

/keep_original_ step	If set, the wavelength output vector will be the same as the one define from the input fits file. The default is to re-construct (and therefore re-inrpolate the galaxy and error spectra) the output wavelength vector with constant ang/pixel step using the minimum ang/pixel step that is stored in the wavelength solution. For MANGA data, it is suggested to set this keyword on.
/use_total	If set, the signal is the total of the counts in the selected wavelength range, the noise is the sum in quadrature of the noises in the selected range. Useful for emission lines.
OUTPUTS	
data	Galaxy spectra. [NxMxT elements array] in the case that the inputs are in datacube format, or [NxT elements array] in the case the inputs are in RSS format. Spectra are resampled over the vector wavelength.
error	Errors associated to data. [NxMxT elements array] in the case that the inputs are in datacube format, or [NxT elements array] in the case the inputs are in RSS format. Spectra are resampled over the vector wavelength.
wavelength	[T elements array]. Wavalenght value (in angstrom) where data and errors are computed. The vector is constructed with constant linear step in ang/pixel (unless the /keep_original_ step keyword is selected). If input spectra have a logarithmic sampling, the minimum available step is used (e.g. log_lam[1]-log_lam[0], converted into angstrom). In the case of MANGA data, we recommend to set the /keep_original_ step keyword, to have the wavelength output vector equal to the wavelength vector of the data.
x2d	Array containing the x coordinates in arcseconds (0 is the center of the field of view). [NxM elements array] in the case that the inputs are in datacube format, or [N elements array] in the case the inputs are in RSS format.
y2d	Array containing the y coordinates in arcseconds (0 is the center of the field of view). [NxM elements array] in the case that the inputs are in datacube format, or [N elements array] in the case the inputs are in RSS format.
signal	Mean galaxy signal per Å, obtained considering all the wavelength range (or only the range specified by lrange), for each input spectrum. [NxM elements array] in the case that the inputs are in datacube format, or [N elements array] in the case the inputs are in RSS format. The signal is computed as the median of each spectrum, in the wavelength range specified by lrange (unless the keyword /use_total is set).

noise	Mean galaxy error per Å, obtained considering all the wavelength range (or only the range specified by lrange)), for each input spectrum. Calculation is done on original spectra, not those resampled over the vector wavelenght. [NxM elements array] in the case that the inputs are in datacube format, or [N elements array] in the case the inputs are in RSS format. The signal is computed as the median of each error, in the wavelength range specified by lrange (unless the keyword /use_total is set).
cdelt1	[double]. Spatial sampling along x direction (arcsec/pixel). If inputs are in RSS format, it is set to 0.5 arcsec/pixel.
cdelt2	[double]. Spatial sampling along y direction (arcsec/pixel). If inputs are in RSS format, it is set to 0.5 arcsec/pixel.
header2d	[str array]. The header for output two-dimensional maps produced by the DAP.
OPTIONAL OUTPUTS	
x2d_ reconstructed	[NxM] elements array] if input is in DATACUBE format, and x2d_ reconstructed = x2d. [N'xM'] elements array] if inputs are in RSS format, the x2d coordinates are resampled over a 2D map with fixed 0"5/pixel sampling and define the x2d_ reconstructed map.
y2d_ reconstructed	[NxM] elements array if input is in DATACUBE format, and y2d_reconstructed = y2d. [N'xM'] elements array if inputs are in RSS format, the y2d coordinates are resampled over a 2D map with fixed 0"5/pixel sampling and define the y2d_ reconstructed map.
signal2d_ reconstructed	[NxM] elements array if input is in DATACUBE format, and signal2d_reconstructed = signal. [N'xM'] elements array if inputs are in RSS format, the signal is resampled over the 2D map defined by x2d_ reconstructed and y2d_ reconstructed.
noise2d_ reconstructed	[NxM] elements array if input is in DATACUBE format, and noise2d_reconstructed = noise. [N'xM' array] If inputs are in RSS format, the noise is resampled over the 2D map defined by x2d_reconstructed and y2d_reconstructed.
version	[string] Module version. If requested, the module is not execute and only version flag is returned.

Note: NaN values in each error vector will be replaced by the median error value (computed using error's defined values).

Future developments

The following items needs to be implemented:

- Implement with new format of input datacube (i.e. read and handlemask arrays).

- Allow for identification and removal of foreground stars.
- Get Milky Way extinction, and provide extinction-corrected galaxy spectra and errors. Insert an appropriate main module (i.e. dust_getval.pro) to do this

2.2 DAP Block 2: Spatial binning

This block adds adjacent spectra in the field of view to reach a minimum signa-to-noise ratio, and organizes the observations into spatial bins.

It requires the spectra, errors, signals and noise for each spectrum, and the coordinates obtained from block 1 as inputs. It returns the binned spectra (and errors) and all the geometrical information of the spatial bin (2D maps and coordinates of the bins center). It handles both datacubes and RSS input files format.

Three different spatial binning schemes with different are handled, depending on the scientific purposes.

1. First binning, which requires a minimum $S/N = 40$ per Å. The spectra binned this way will be used to measure the equivalent width ob absorption line indices, to perform the full spectral fitting, and to derive the stellar population properties.

2. Second binning, which requires a minimum $S/N = 25$ per Å. The spectra binned this way will be used to measure the stellar kinematics and dynamical models.

3. Third binning, which requires a minimum $S/N = 15$ per Å. The spectra binned this way will be used to measure the emission line kinematics, fluxes, and widths, the reddening, gas metallicities and star formation rates.

S/Ns are computed in the wavelength range $5560 < \lambda < 6942$ (SLOAN r-band).

The interface module in this block is mdap_spatial_binning.pro (see Section 2.2.1). The main module is mdap_voronoi_2d_binning.pro (Section 3.3).

mdap_voronoi_2d_binning.pro uses an imlementation of the Voronoi Binning technique developed by **?** (see Section 3.3). The implementation accounts for spatial correlation between errors of adjacent spectra: the estimated S/N is calibrated to the real S/N via empirical relation and it is done by the utility module mdap_calibrate_sn.pro.

Note that RSS spectra do not require the above S/N calibration, because the errors between adjacent RSS files are not correlated.

The interface mdap_spatial_binning.pro accepts an user-defined fits file describing the spatial binning scheme. This file needs to be provided via the input keyword: user_bin_map. The names are defined in the DAP configuration file (see Section 1.3). The inclusion of an used-defined file overrides the Voronoi binning.

On request, the spectra in the same spatial bin can be added together by weighting them with their S/N^2 (weight_by_sn keyword).

Figure 2.2 illustrates the flowchart of block 2.

Future implementations

Future implementations:

- calculation of spatial covariance matrix to account for error correlations and avoid the use of S/N calibration relations.

- Allow for an equal flux surface binning scheme (Sanchez et al. ?) instead of the Voronoi binning scheme.

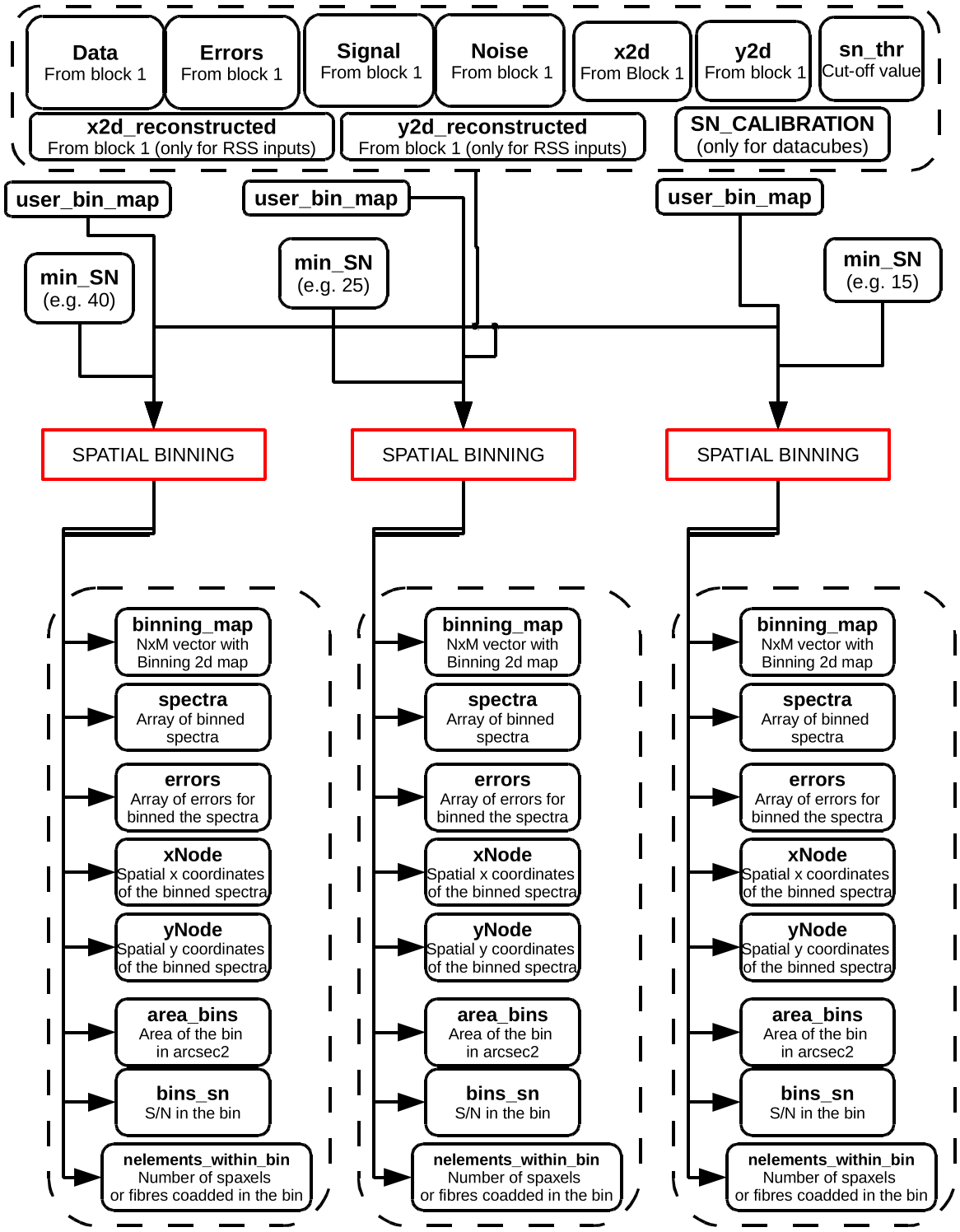

Figure 2.2: Workflow chart of the block 2 of the Data Analysis Pipeline.

2.2.1 `mdap_spatial_binning.pro`

This module is an interface for the mdap_voronoi_2d_binning.pro module, which uses the Voronoi Binning scheme (modified Loyd algorithm) to combine adjacent spectra in the field of view till a minimum S/N is reached (see Section 3.3).

If the minimum S/N is not reached, the requirement will be automatically decreased by a factor 0.7 till convergence. By default, spectra with mean $S/N < 4$ per Åwill be discarded. At least, 1 spectrum per datacube will be produced.

If user-defined spatial binning is provided (via the user_bin_map keyword) the Voronoi binning scheme is overrided.

Table 2.2 lists the inputs and outputs required for this module.

Table 2.2: Inputs and outputs parameters of mdap_spatial_binning.pro

INPUTS	
data	Galaxy spectra as produced by mdap_read_datacube.pro. [NxMxT dbl array] if DATACUBE format or [NxT dbl array]if RSS format.
error	Errors associated to data, produced by mdap_read_datacube.pro. [NxMxT dbl array] if DATACUBE format or [NxT dbl array]if RSS format.
signal	Mean galaxy signal per Å, produced by mdap_read_datacube.pro. [NxM dbl array] if DATACUBE format or [N dbl array]if RSS format.
noise	Mean galaxy error per Å, produced by mdap_read_datacube.pro. [NxM dbl array] if DATACUBE format or [N dbl array]if RSS format.
min_sn	[float] Minimum S/N (per Å) required for the output binned spectra.
x2d	Array containing the x coordinates in arcseconds (0 is the center of the field of view), produced by mdap_read_datacube.pro. [NxM dbl array] if DATACUBE format or [N dbl array]if RSS format.
y2d	Array containing the y coordinates in arcseconds (0 is the center of the field of view), produced by mdap_read_datacube.pro. [NxM dbl array] if DATACUBE format or [N dbl array] if RSS format.
stepx	[float] Scale arcsec/pixel along X direction, computed by mdap_read_datacube.pro.
stepy	[float] Scale arcsec/pixel along Y direction, computed by mdap_read_datacube.pro.
OPTIONAL INPUTS	
sn_thr	[float] If specified, spectra with S/N lower than this value will be excluded from the analysis.
x2d_ reconstructed	[N'xM' array] Two-dimesional map of X coordinates where the output spatial_binning should be created. Required and used only if the input data are in RSS format.

y2d_ reconstructed	[N'xM' array] Two-dimesional map of Y coordinates where the output spatial_binning should be created. Required and used only if the input data are in RSS format.
SN_ CALIBRATION	vector. If provided, the estimated signal-to-noise (`SN_est`) is converted into the real signal-to-noise (`SN_real`) using the empirical calibration function defined in mdap_calibrate_sn.pro: $$\mathrm{S/N_{REAL}} = \sum_{i=1,N} C_i \cdot \left(\frac{\mathrm{S/N_{ESTIMATED}}^{C_0}}{\sqrt{\mathrm{N_{spax}}}} \right)^{i-1}$$ where `Nbin` is the number of spectra added in that spatial bin.
user_bin_map	string If provided, the spatial map will be created from the fits fiel specified by this input. The fits file must contain the CRVAL1, CRVAL2, CDELT1, CDELT2, NAXIS1, NAXIS2, CRPIX1, and CRIX2 header keywords (coordinate units in arcseconds, 0,0 indicates the center of the field of view.
INPUT KEYWORDS	
/plot	If set, some plots on X11 terminal will be shown. Not suggested if the task is launched remotely.
/weight_for_sn	If set, the spectra in the same spatial bin will be weighted by S/N^2 before being added. If The voronoi binning scheme is adopted, the S/N in the bin is computed via equation (3) of Cappellari & Copin (2003) and the centers of te spatial bins are computed by weighting spectra coordinates by S/N^2.
OUTPUTS	
binning_map	Two dimensional map showing the binning sheme. Pixels beloning to the i-th bin have value i (i=0, 1, ..., Nbins). Pixels associated to no spatial bin have value -1. [NxM dbl array] if inputs are in DATACUBE format or [N'xM' dbl array] if inputs are in RSS format (interpolated over x2d_reconstructed and y2d_reconstructed).
spectra	[Nbins x T dbl array] The binned spectra of the spatial Nbins. i-th spectrum is associated to the i-th bin.
errors	[Nbins x T dbl array] Errors vectors associate do the binned spectra.
xNode	[Nbins elements array] X-Coordinates in arcsec of the luminosity weighted centers of the spatial bins.
yNode	[Nbins elements array] Y-Coordinates in arcsec of the luminosity weighted centers of the spatial bins.
area_bins	[Nbins elements array] Area (in arcsec^2) of each spatial bin.
bin_sn	[Nbins elements array] Mean S/N per Åreached in each spatial bin.
OPTIONAL OUTPUTS	
nelements_ within_ bin	[Nbins elements array] number of spaxels (in the case of DATACUBE format) or number of fibres (in the case of RSS format) coadded in each spatial bin.

version	[string] Module version. If requested, the module is not execute and only version flag is returned.

2.3 DAP Block 3: Logarithmic sampling of input galaxy spectra and stellar templates

This block is responsable for:

- resampling the input galaxy spectra and errors (rebinned spectra, input from block 2) to a ln-wavelength constant step.
- resampling the stars in the spectral library in the same ln-wavelength constant step used for galaxy spectra, and ensuring that the stars have the same instrumental resolution (i.e. the same LSF) of the galaxy spectra.
- selecting only the wavelength region of interest, making sure that the wavelenght range of the template stars is larger than that of the galaxy spectra.
- defyning the wavelenght vectors (ln units) over which the galaxy spectra, galaxy errors, and template stars are defined.

Because the DAP performs three different spatial binnings of the input galaxy, the block operates

The interface module in this block is mdap_log_rebin.pro (see Section 2.3.1), which calls the main module mdap_do_log_rebin.pro that performs the logarithmic resampling (see Section 3.2). Because the DAP performs three different spatial binnings of the input galaxy, block 3 executes mdap_log_rebin.pro three times, one for each set of spatially binned spectra. The workflow of block 3 is shown in Figure 2.3.

The current version of the DAP uses the MARCS stellar library models (Maraston & Stromback, 2011, MNRAS, 418, 2785).

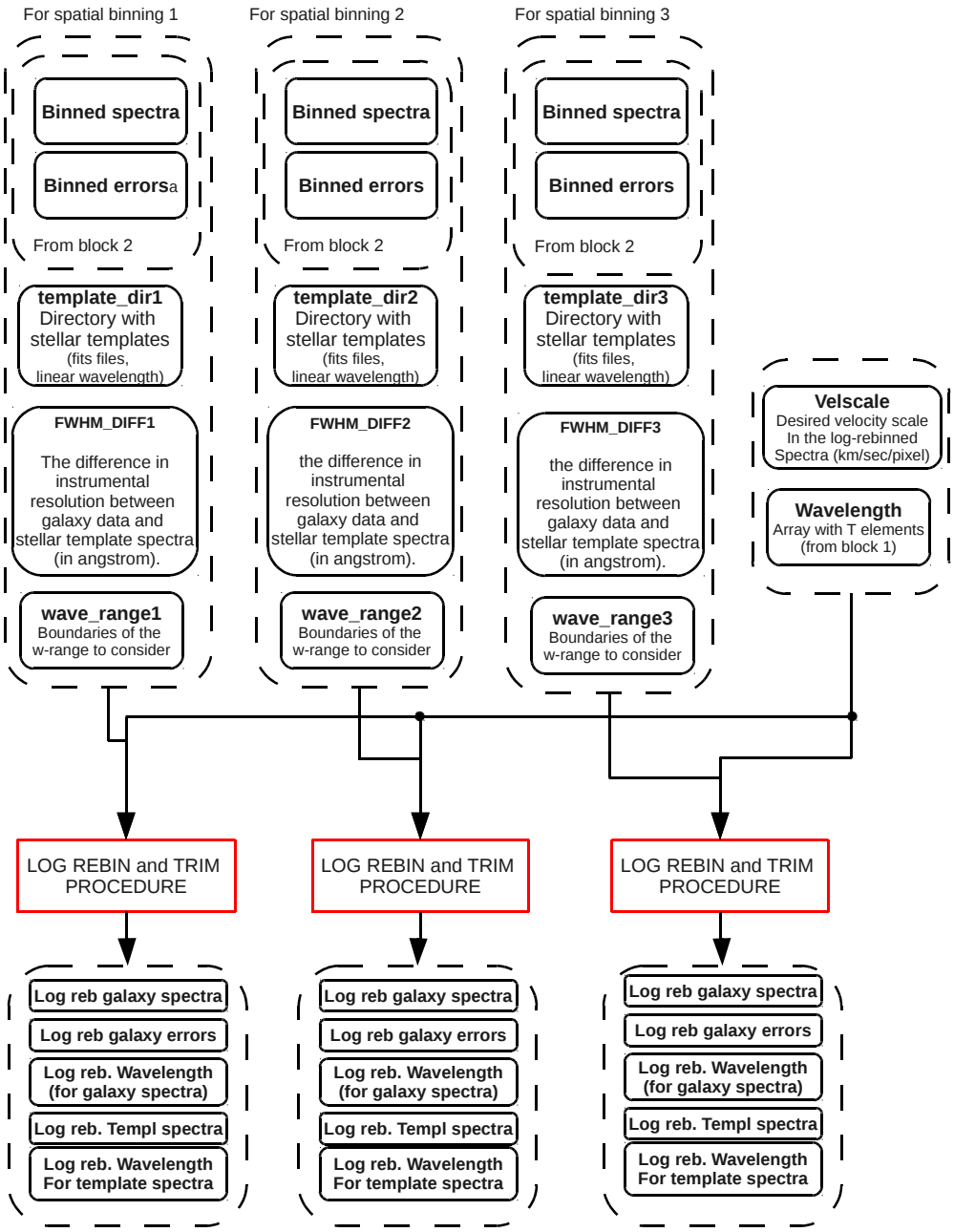

Figure 2.3: Workflow chart of the block 3 of the Data Analysis Pipeline.

2.3.1 `mdap_log_rebin.pro`

This interface controls the main module is responsable for the logarithmic resampling of the input galaxy spexctra, errors, and the stellar templates. It also broadens the input stellar spectra to match the galaxy instrumental resolution.

The list of input/output parameters defined in mdap_log_rebin.pro is given in Table 2.3.

Table 2.3: Inputs and outputs parameters of mdap_log_rebin.pro

INPUTS	
spectra	[NxT dbl array]. The N galaxy spectra to resample.
errors	[NxT dbl array]. Errors associated to the N spectra.
wavelength	[T elements array]. Array with T elements, that specifies the wavelengths where galaxy spectra and errors are defined.
library	[string]. Directory containing the template spectra. Template spectra must be in fits format, defined over a linear wavelength with constant ang/pix step, and must contain all the header information needed to recover the wavelenght domain (i.e. CRPIX1, CDELT1, and CRVAL1).
fwhm_diff	[T elements array]. Vector containing the FWHM(λ) (in Å) measured over the vector wavelength that the stellar spectra must be convolved for. It correspond to the difference in quadrature between the instrumental FWHM of the observations, and the instrumental FWHM of the templates. The convolution is done with the utility module mdap_convo_sigma.pro (see Section 4.1).
OPTIONAL INPUTS	
input_velscale	[flt]. Constant km/sec/pixel to be used when rebinning the input spectra. If not provided, the value will be automatically set by the procedure.
wave_range	[2 elem array]. If specified, the galaxy spectra will be trimmed to this wavelength range (units: angstrom). Default: use the entire input wavelength range. Stellar spectra will be trimmed by wave_range[0] - 250 ang and wave_range[1] + 250 ang.
OPTIONAL KEYWORDS	
/flux	If set, flux conservation is applied to the log resampling. **Do not use** for template fitting. Suggestion: do not use.
/gal_wavelength_log_step	Set this keyword if the input galaxy spectra are logarithmically sampled (i.e. wavelength has a logarithmic progression). Suggested for MaNGA data.
/quiet	If set, message prompt is suppressed.
OUTPUTS	
log_spc	[N x TT dbl array]. The logarithmically resampled (ln-lambda) N galaxy spectra, over the wavelength range ln(wave_range).

log_err	[N x TT dbl array] The errors associated to the log_spc. Errors are rebinned using the following formulas: lrg=minmax(wavelength) mdap_do_ log_rebin, lrg, errors^2, log_err2, loglam, velscale=velscale log_err = $\sqrt{\text{log\_err2}}$ where mdap_do_log_rebin.pro is the original procedure by M. Cappellari (see Section 3.2).
log_wav	[TT dlb array]. The values of the ln-lambda over which log_spc and log_err are defined.
library_log	[W x M dbl array]. The W stellar template spectra, logarithmically resampled.
log_wav_ library	[M dbl array]. The values of log-lambda over which the stellar templates are defined.
OPTIONAL OUTPUTS	
version	string specifying the module version. If requested, the module is not execute and only version flag is returned.

2.4 DAP Block 4: Spectral fitting

This block is responsible for fitting the input galaxy spectra with a set of stellar templates and Gaussian Emission lines to derive the kinematics of stars, ionized gas, and emission line fluxes and equivalent widths.

This block has two interfaces: mdap_spectral_fitting.pro (see Section 2.4.2) and mdap_create_starting_guesses (see Section 2.4.1).

The interface mdap_create_starting_guesses handles the starting guesses (either from the user input file, or using the results of a previous fit) and feeds the spectral fitting interface.

The interface mdap_spectral_fitting arranges inputs and outputs for the routine that performs the actual fit, i.e. mdap_gandalf_wrap.pro, which is an adapted version of the pPXF (**?**) and gandalf (**?**) routines (see Section **??** for information on how the fit is performed).

Block 4 requires as input the log-resampled spectra of the galaxy (and errors) and the stars, wavelenght vectors, the binning spatial informations, and starting guesses for the galaxy redshift and velocity dispersion.

The fitting procedure can be fed with a vector that describes the variation of the instrumental velocity dispersion with wavelength [1]. If not provided, a constant instrumental velocity dispersion of 52 km/sec will be adopted.

The block executes the module mdap_spectral_fitting.pro three times, once for each set of spatially binned spectra. The current version of the DAP uses the MARCS library of ssp (Maraston & Stromback, 2011, MNRAS, 418, 2785) for the spectral fitting.

1. *First module execution.* The log-sampled Galaxy spectra are fitted with a linear combination of templates and Gaussian Emission lines. This step requires input starting guesses of the galaxy redshift and stellar velocity dispersion (more than one redshift, and coordinates of galaxy centers if more galaxies are presented in the field of view). The galaxy spectra with the best-fit emission lines removed produced as output will be used to measure the absorption line strenghts (block 5). Galaxy reddening is also fitted, if required. This execution does not provide final highlevel data products, but it will provides: i) galaxy spectra with the best-fit emission lines removed (which will be used in block 5 to measure the absorption line indices), ii) the weights of the stellar templates (which will be used to compute the stellar populations in block 7).

2. *Second module execution.* The results of the first execution (stellar and emission lines kinematics) are re-sampled over the second spatial sampling and used as starting guesses for the second execution on the second set of galaxy spectra (second spatial binning). This step is performed by the interface mdap_create_starting_guesses.pro (Section 2.4.1) and the utility mdap_interpolate_2dmaps.pro (Section 4.3. As in the first execution, log-sampled Galaxy spectra are fitted with a linear combination of stellar templates and Gaussian Emission lines. Only those stellar templates that are selected in the first execution, will be used in the second run (unless the keyword /dont_remove_ null_templates of the manga_dap is set, see Section 1.2.1).

3. *Third module execution.* The emission line kinematics from the second execution are re-sampled over the third spatial sampling and used as starting guesses for the third execution. The stellar kinematics from the second execution are re-sampled over the third

[1] This affects only emission lines measurements, because the stellar templates have already been convolved in Block 3 to match the instrumental resolution, see 2.3

spatial sampling and *fixed* in the third execution. This step is performed by the interface mdap_create_starting_guesses.pro (Section 2.4.1) and the utility mdap_interpolate_2dmaps.pro (Section 4.3. As in the first execution, log-sampled Galaxy spectra are fitted with a linear combination of stellar templates and Gaussian Emission lines. Only those stellar templates that are selected in the first execution, will be used in the third run (unless the keyword /dont_remove_ null_templates of the manga_dap is set, see Section 1.2.1).

At the end of block 4, the stellar kinematics (V, σ, h3, and h4), emission line kinematics (V, σ), emission line fluxes and equivalent widths, reddening (stars and gas) and stellar template weights for each of spectrum and for the spatial binning scheme are produced.

Figures 2.4 and **??** shows the flowchart of block 4.

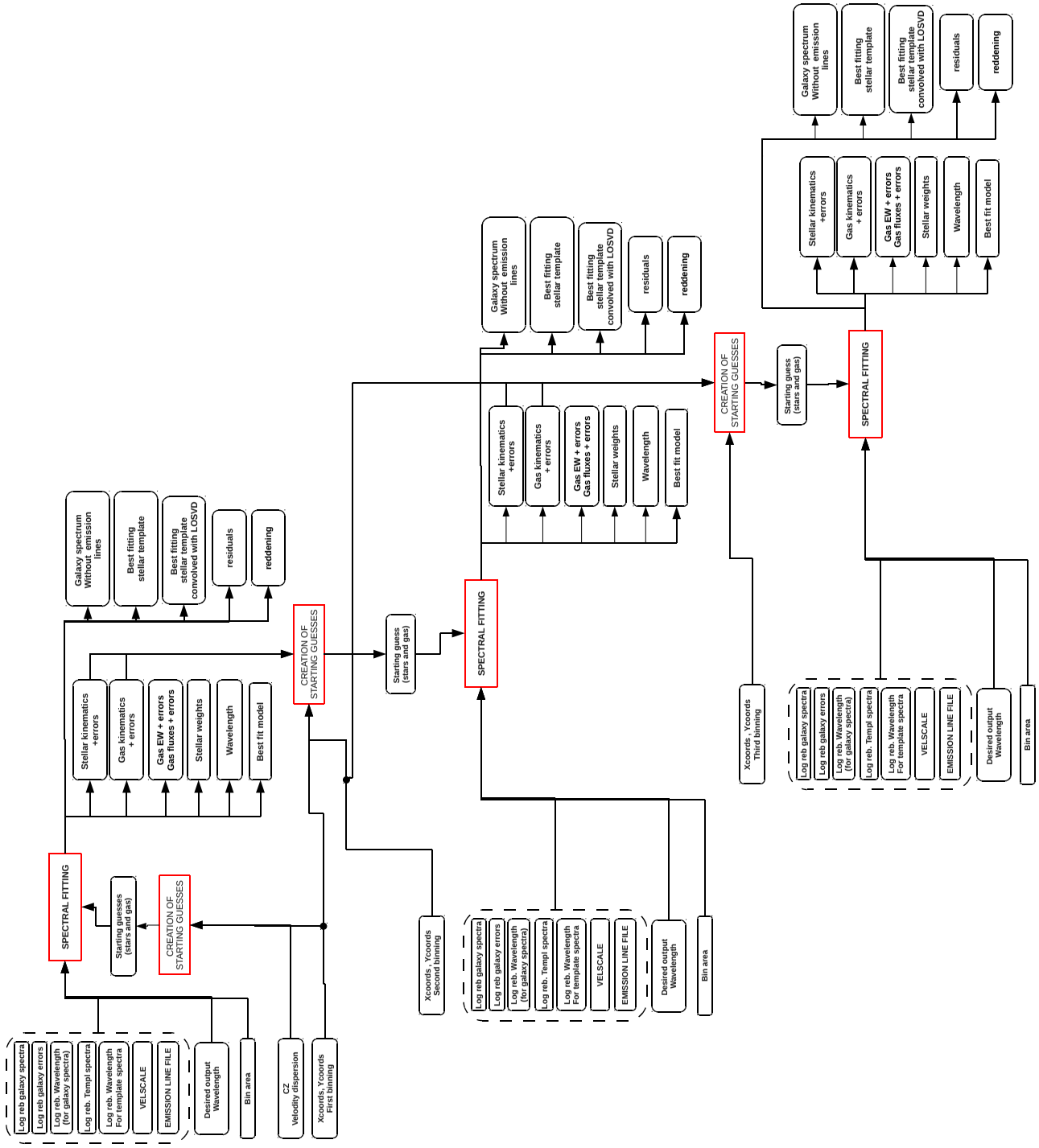

Figure 2.4: Workflow chart of the block 4 of the Data Analysis Pipeline.

2.4.1 `mdap_create_starting_guesses.pro`

This interface is responsible to transform the kinematics measured on a spatial binning scheme into starting guesses for the fit of another spatial binning scheme. The main module that performs the computation is mdap_interpolate_2dmaps.pro (see Section 4.3).

Table 2.4 lists the inputs and outputs of mdap_create_starting_guesses.pro.

Table 2.4: Inputs and outputs parameters of mdap_create_starting_guesses.pro

INPUTS	
input_map	Nx4 (or Nx2) elements array containing the measurements of velocity, velocity dispersion of the N spectra to be used to get the starting guesses. If the keyword /h3h4 is set, input_map must be Nx4 elements array, and previous measurements of h3 and h4 must be given. input_map[*,0]: previous velocity measurements input_map[*,1]: previous velocity dispersion measurements input_map[*,2]: (optional) previous H3 measurements input_map[*,3]: (optional) previous H4 measurements
input_xbin	N elements array. X coordinates of the center of the spatial bins where the measurements in input_map refer to.
input_ybin	N elements array. Y coordinates of the center of the spatial bins where the measurements in input_map refer to.
x2d	N'xM' elements array. Two dimensional map of X coordinates of a regular grid mapping the full field of view.
y2d	N'xM' elements array. Two dimensional map of Y coordinates of a regular grid mapping the full field of view.
velocity_initial_guess	Float. Guess for the velocity to be used in the case of previous measurements are NaN, or not defined.
velocity_dispersion_initial_guess	Float. Guess for the velocity to be used in the case of previous measurements are NaN, or not defined, or outside a fiducial interval $21 < \sigma < 499$ km/sec.
H3_initial_guess	Float. Guess for the velocity to be used in the case of previous measurements are NaN, or not defined, or outside a fiducial interval $-0.399 < h3 < 0.399$.
H4_initial_guess	Float. Guess for the velocity to be used in the case of previous measurements are NaN, or not defined, or outside a fiducial interval $-0.399 < h4 < 0.399$.
output_xbin	M elements array. X coordinates of the center of the spatial bins where the starting guesses are to be computed.
output_ybin	M elements array. Y coordinates of the center of the spatial bins where the starting guesses are to be computed.
KEYWORD	

/h3h4	If set, the guesses of Gauss Hermite moments (h3, h4) will be computed (otherwise set to H3_initial_guess and H4_initial_guess) . The input input_map must contain h3 and h4 measurements.
OUTPUTS	
output_start_guess	Mx4 elements array containing the interpolated velues of of velocity, velocity dispersion of the N spectra to be used as starting guesses. If the keyword /h3h4 is set, it will be Mx4 elements array. output_start_guess[*,0]: velocity interpolated on the output_xbin, output_ybin, grid to be used as starting guesses. output_start_guess[*,1]: velocity dispersion interpolated on the output_xbin, output_ybin, grid to be used as starting guesses. output_start_guess[*,2]: (optional) H3 interpolated on the output_xbin, output_ybin, grid to be used as starting guesses. output_start_guess[*,3]: (optional) H4 interpolated on the output_xbin, output_ybin, grid to be used as starting guesses.

2.4.2 mdap_spectral_fitting.pro

This module is responsable for i) arranging the inputs from previous modules/blocks and feed them into the main module that perform the fit (mdap_gandalf_wrap.pro, see Section **??**) , and ii) re-arrange the outputs of the fits for further processing.

The list of input/output parameters of mdap_spectral_fitting.pro is given in Table 2.5.

Table 2.5: Inputs and outputs parameters of mdap_spectral_fitting.pro

INPUTS	
galaxy	[MxN array]. It contains the N galaxy spectra to fit, logarithmically sampled (natural log). Units: 1e-17 erg/s/cm^2/Angstrom. Spectra are defined of the M elements array loglam_gal (see below).
noise	[MxN array]. It contains the N error vectors for the N galaxy spectra. Same units as galaxy. Error vectors are defined of the M elements array loglam_gal (see below).
loglam_gal	[M array]. It contains the log wavelength values where galaxy and noise spectra are defined. The vector must have a constant log(angstrom) sampling.
templates	[MM x NN array]. It contains the NN stellar template spectra, logarithmically sampled at the same km s^{-1}/pixel as the galaxy spectra. Same units as galaxy, except an arbitrary multiplicative factor.
loglam_templates	[MM dblarray]. It contains the log wavelength values where templates are sampled. It must have a constant log(angstrom) sampling.
velscale	[float]. Defines the (uniform) sampling of the input spectra, in km s^{-1}/pixel.
OPTIONAL INPUTS	

extra_inputs	[string array] It contains other inputs that might be used in the fitting procedure, such as the number of polinomyal degree. Variable will be initialized with the IDL execute command.
	`for i = 0, n_elements(extra_inputs)-1 do d = execute(extra_inputs[i])`
	EXAMPLE:
	`extra_inputs=['MOMENTS=2','DEGREE=-1','MDEGREE=3', 'BIAS=0.2', 'reddening=[0.1,0]', 'LAMBDA=exp(loglam_gal)'].`
	Warning: The reddening (stars and/or gas) fit is optional, and it is performed by the Gandalf module. If the reddening fit is required, MDEGREE and DEGREE are used as the input value for the pPXF run, but automatically set to 0 and -1 respectively for the Gandalf execution.

star_kin_starting_guesses	[N x 4 fltarray]. The stellar kinematics starting guesses for V, σ, H3, and H4 for the N galaxy spectra to fit. Default values are 0. for V, H3, and H4, and 50 km/sec for sigma. Starting guesses values are overrided by the /use_ previos_ guesses keyword, if set.
gas_kin_starting_guesses	[N x 2 fltarray]. The emission line kinematics starting guesses for V, σ, for the N galaxy spectra to fit. Default values are 0 km s^{-1} for V, and 50 km s^{-1} for sigma. Starting guesses values are overrided by the /use_previos_guesses keyword, if set.

emission_line_file	[string]. It contains the name of the file with the information of the emission lines to be fitted. The input file must be an ascii file with the following columns (comments starts with "#"):
	`# ID CODE wav action line Int Vel σ mode`
	`# [Å] f/i/m dbl?`
	`# 0 HeII 3203.15 m l 1.0 0 10 t25`
	`# 1 [NeV] 3345.81 m l 1.0 0 10 t25`
	`2 [NeV] 3425.81 m l 1.0 0 10 t25`
	`3 [OII] 3726.03 m l 1.0 0 10 t25`

range_v_star	[2 elements array]. It specifies the boundaries for the stellar best fit velocity (in km s^{-1}). Default: starting_guess $\pm$ 2000 km s^{-1}.
range_s_star	[2 elements array]. It specifies the boundaries for the stellar best fit velocity dispersion (in km s^{-1}). Default: $21 < \sigma < 499$ km s^{-1}.
range_v_gas	[2 elements array]. It specifies the boundaries for the emission line best fit velocity (in km s^{-1}) Default: starting_guess $\pm$ 2000 km s^{-1}.
range_s_gas	[2 elements array]. It specifies the boundaries for the emission line best fit velocity dispersion (in km s^{-1}). Default: starting_guess $\pm$ 2000km s^{-1}.
wavelength_input	[QQ elements array]. If specified, it will be used to create wavelength_output (in Å), i.e. the wavelength vector to interpolate the final results on. if keyword /rest_frame_log is set, the vector is set to exp(loglam_templates), and user inpiut will be overwritten. In this case QQ = MM. Suggested entry: use the exp(loglam_templates).
external_library	String that specifies the path to the external `FORTRAN` library, which contains the fortran versions of mdap_bvls.pro. If not specified, or if the path is invalid, the default internal IDL mdap_bvls code is used.
mask_range	If defined, it specifies the wavelength ranges to mask in the fit. It must contain an even number of entries, in angstrom. E.g. mask_range=[λ_0, λ_1, λ_2, λ_3, ...$\lambda_{(2n-1)}$, $\lambda_{(2n)}$] will mask all the pixels where the $\lambda_0 < exp$(loglam_gal) $< \lambda_1$; $\lambda_2 < exp$(loglam_gal) $< \lambda_3$; $\lambda_{(2n-1)} < exp$(loglam_gal) $< \lambda_{(2n)}$.

fwhm_instr_ kmsec_matrix	2xW elements vector. It defines the instrumental FWHM as function of wavelength. fwhm_instr_ kmsec_matrix[0,*] specifies the wavelength (in angstrom, at which the instrumental FWHM is measured. fwhm_instr_ kmsec_matrix[1,*] specifies values of the instrumental FWHM (in km/sec) measured at fwhm_instr_ kmsec_matrix[0,*]. If undefined, a constant instrumental FWHM=122.45 km/sec is adopted.
OPTIONAL KEYWORDS	
/use_previos_ guesses	If set, the starting guesses for spectrum $i-$th will be the best fit values from spectrum $(i-1)-$th (i$>$ 0). Input starting guesses will be ignored.
/fix_star_kin	If set, the stellar kinematics are not fitted. The return values is that of the starting guesses.
/fix_gas_kin	If set, the emission-lines kinematics are not fitted. The return values is that of the starting guesses.
/quiet	If set, some information are not printed on screen.
/rest_ frame_log	If set, the output spectra (galaxy_minus_ ems_fit_model, best_fit_model, residuals, best_template, and best_template_LOSVD_conv) are produced at rest-frame wavelength.
OUTPUTS	
stellar_ kinematics	[N x 5 flt array]. It contains the best fit values of V, σ, h3, h4, and χ^2/DOF for each of the N fitted input galaxy spectra. If /fix_star_kin is set, the array is not defined.
stellar_ kinematics_err	[N x 5 flt arrary]. It contains the errors to the best fit values of V, σ, h3, h4, and χ^2/DOF for each of the N fitted input galaxy spectra.
stellar_weights	[N x NN dbl array]. It contains the weights of the NN templates for each of the N input galaxy spectra.
emission_line_ kinematics	[N x 2 flt array]. It contains the best fit values of V, σ (emission lines) for each of the N fitted input galaxy spectra. If /fix_gas_kin is set, the array is not defined.
emission_line_ kinematics_err	[N x 2 flt array]. It contains the errors to the best fit values of V, σ (emission lines) for each of the N fitted input galaxy spectra. If /fix_gas_kin is set, the array is not defined.
emission_line_ fluxes	[N x T flt array]. It contains the fluxes of the T fitted emission lines for each of the N input galaxy spectra. Values are corrected for reddening.
emission_line_ fluxes_err	[N x T flt array]. Errors associated to emission_line_fluxes
emission_line_ equivW	[N x T flt array]. It contains the Equivalent widths of the T fitted emission lines for each of the N input galaxy spectra. Equivalent widths are computed by the ratio of emission_line_fluxes and the median value of the stellar spectrum within 5 and 10 σ from the emission line. σ is the emission line velocity dispersion.

emission_line_ equivW_err	[N x T flt array]. Errors associated to emission_line_equivW .
wavelength_ output	[QQ elements flt array]. It will contain the wavelength values over which the output spectra are sampled (in Å). Default: it is set to wavelength_input (if defined), or automatically computed with the smallest lambda/pixel step obtained from exp(loglam_gal). It is set to exp(LOGLAM_TEMPLATES) if the keyword /rest_ frame_log is set.
best_fit_model	[N x QQ flt array]. It will contain the best fit models for each of the input galaxy spectra (dereddended if MW_extinction is not zero), sampled over wavelength_output. It is in rest-frame if the keyword /rest_frame_log is set.
galaxy_minus_ ems_fit_model	[N x QQ flt array]. It will contain the input galaxy spectra minus the emission lines best fit models (dereddended if MW_extinction is not zero), sampled over wavelength_output, for each of the N input spectra. It is in rest-frame if the keyword /rest_frame_log is set
best_template	[N x QQ flt array]. It will contain the best fitting template for each of the N input galaxy spectra sampled over wavelength_output (rest frame wavelength).
best_template_ LOSVD_conv	[N x QQ flt array]. It will contain the best fitting template for each of the N input galaxy spectra convolved by best fitting LOSVD and sampled over wavelength_output (rest frame wavelength).
reddening_ output	[float]. best fit value for the reddening, if the fit is required (otherwise the variable is not defined). To fit the reddening, you have to pass a starting guess value and the LAMBDA = exp(loglam_gal) vector through the extra_keyword parameter. Example: extra_inputs = ['reddening=0', 'LAMBDA=exp(loglam_gal)'].
residuals	[N x QQ flt array]. It contains the difference between the observed galaxy spectra (dereddened if the MW_reddening is defined) and the best_fit_model, sampled over wavelength_output. It is in rest-frame if the keyword /rest_frame_log is set
OPTIONAL OUTPUTS	
version	string specifying the module version. If requested, the module is not execute and only version flag is returned.

2.5 DAP Block 5: Measurement of absorption line indices

This module is responsible for measuring the equivalent width of absorption line indices on the galaxy spectra where the best fit model for emission lines has been removed.

It requires the following inputs:

- The galaxy spectra with emission lines removed (from block 4).
- The stellar velocity.
- The LSF as function of wavelenght (to a proper broadening of the input spectra to the reference calibration system, e.g. Lick, MILES).
- The best fitting stellar template and the best fitting stellar template convolved by the best fitting LOSVD, for intrinsic broadening correction (from block 4).

Figure 2.5 illustrates the flowchart of this block.

The main module in this block is mdap_measure_ indices.pro (see Section 3.3.2), which calls the procedure that performs the measurement mdap_do_ measure_index.pro (see Section ??).

Figure 2.5 illustrates the flowchart of the main module in this block.

2.6 DAP Block 6: Radial properties of high level data products

2.7 DAP Block 7: Stellar population analysis

T.B.D.

2.8 DAP Block 8: Properties of the emission lines

T.B.D.

2.9 DAP Block 9: Mass modeling

T.B.D.

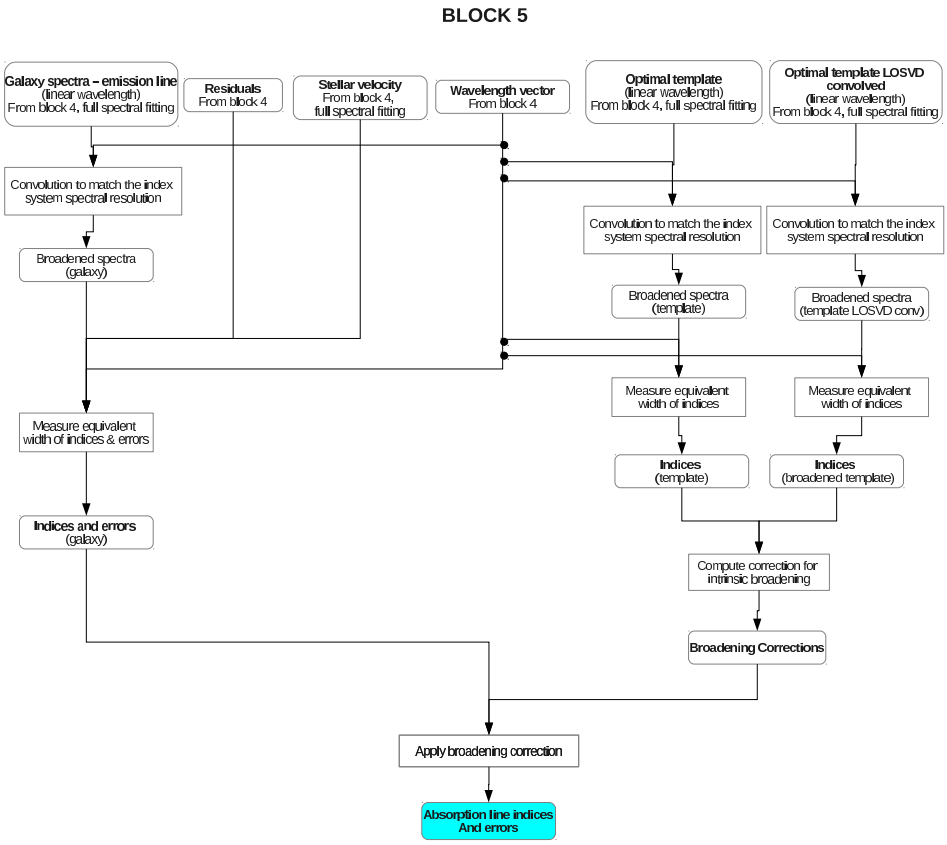

Figure 2.5: Workflow chart of mdap_measure_indices.pro, the main module in block 5 of the Data Analysis Pipeline.

Chapter 3

Main modules

In this Chapter we describe the main modules of the DAP.

3.1 mdap_calculate_spectrum_sn.pro

bla bla bla

3.2 mdap_do_log_rebin.pro

This module is called by the mdap_log_rebin.pro interface and performs the actual logarithmic resampling. It has been originally written by M. Cappellari within the ppxf package. Here we report the original description of the procedure written by M. Cappellari.

NAME: MDAP_DO_LOG_REBIN

PURPOSE: Logarithmically rebin a spectrum, while rigorously conserving the flux (if the keyword \flux is set). Basically the photons in the spectrum are simply ridistributed according to a new grid of pixels, with non-uniform size in the spectral direction.

This routine makes the 'standard' zero-order assumption that the spectrum is *constant* within each pixels. It is possible to perform log-rebinning by assuming the spectrum is represented by a piece-wise polynomial of higer degree, while still obtaining a uniquely defined linear problem, but this reduces to a deconvolution and amplifies noise.

This same routine can be used to compute approximate errors of the log-rebinned spectrum. To do this type the command:

$$\text{MDAP\_DO\_LOG\_REBIN}, \text{lamRange}, \text{err}^2, \text{err2New}$$

and the desired errors will be given by SQRT(err2New). NB: This rebinning of the error-spectrum is very *approximate* as it does not consider the correlation introduced by the rebinning!

The list of inputs/outputs of the mdap_do_log_rebin.pro is given in Table 3.1.

Table 3.1: Inputs and outputs parameters of mdap_do_log_rebin.pro

INPUTS	
LAMRANGE	two elements vector containing the central wavelength of the first and last pixels in the spectrum, which is assumed to have constant wavelength scale! E.g. from the values in the standard FITS keywords: LAMRANGE = CRVAL1 + [0, CDELT1 * (NAXIS1-1)]. It must be LAMRANGE[0] ¡ LAMRANGE[1].
SPEC:	N elements array. The input spectrum.
OUTPUTS	
SPECNEW	M elements array. The logarithmically rebinned spectrum.
LOGLAM	log(lambda)M elements array. natural logarithm (ALOG) of the central wavelength of each pixel. This is the log of the geometric mean of the borders of each pixel.
OPTIONAL KEYWORDS	

/FLUX:	Set this keyword to preserve total flux. In this case the log rebinning changes the pixels flux in proportion to their dLam so the following command will show large differences beween the spectral shape before and after LOG_REBIN:
	`plot, exp(logLam), specNew`
	`oplot, range(lamRange[0],lamRange[1],n_elements(spec)), spec`
	By default, when this keyword is *not* set, the above two lines produce two spectra that almost perfectly overlap each other. Do not set /flux for MaNGA data.

OVERSAMPLE:	Oversampling can be done, not to loose spectral resolution, especally for extended wavelength ranges and to avoid aliasing. Default: OVERSAMPLE=1, i.e. Same number of output pixels as input.
VELSCALE:	velocity scale in km/s per pixels. If this variable is not defined, then it will contain in output the velocity scale. If this variable is defined by the user it will be used to set the output number of pixels and wavelength scale.

3.3 mdap_voronoi_2d_binning.pro

This procedure is taken from the Voronoi Binning procedure by Cappellari & Copin (2003). It has been modified so it automatically relaxes the minimal S/N requirements to have at least 1 bin in the field of view. Table 3.2 lists the inputs and outputs required for this module.

Table 3.2: Inputs and outputs parameters of mdap_voronoi_2d_binning

INPUTS	
X	[flt array] Vector containing the X coordinate of the pixels to bin. Arbitrary units can be used (e.g. arcsec or pixels). In what follows the term ?pixel? refers to a given spatial element of the dataset (sometimes called ?spaxel? in the IFS community): it can be an actual pixel of a CCD image, or a spectrum position along the slit of a long-slit spectrograph or in the field of view of an IFS (e.g. a lenslet or a fiber). It is assumed here that pixels are arranged in a regular grid, so that the pixel size is a well defined quantity. The pixel grid however can contain holes (some pixels can be excluded from the binning) and can have an irregular boundary. See the above reference for an example and details.
Y	[flt array]. Vector (same size as X) containing the Y coordinate of the pixels to bin.
SIGNAL	[flt array]. Vector (same size as X) containing the signal associated with each pixel, having coordinates (X,Y). If the 'pixels' are actually the apertures of an integral-field spectrograph, then the signal can be defined as the average flux in the spectral range under study, for each aperture. If pixels are the actual pixels of the CCD in a galaxy image, the signal will be simply the counts in each pixel.
NOISE	[flt array]. Vector (same size as X) containing the corresponding noise (1 sigma error) associated with each pixel.
TARGETSN	[float]. The desired signal-to-noise ratio in the final 2D-binned data. E.g. a S/N$\sim$50 per pixel may be a reasonable value to extract stellar kinematics information from galaxy spectra.
OPTIONAL INPUT :	
SN_ CALIBRATION	vector. If provided, the estimated signal-to-noise (`SN_est`) is converted into the real signal-to-noise (`SN_real`) using the empirical calibration function defined in mdap_calibrate_sn.pro: $$\mathrm{S/N_{REAL}} = \sum_{i=1,N} C_i \cdot \left(\frac{\mathrm{S/N_{ESTIMATED}}^{C_0}}{\sqrt{\mathrm{N_{spax}}}} \right)^{i-1}$$ where `Nbin` is the number of spectra added in that spatial bin.
INPUT KEYWORDS:	

/NO_CVT	Set this keyword to skip the Centroidal Voronoi Tessellation (CVT) step (vii) of the algorithm in Section 5.1 of Cappellari & Copin (2003). This may be useful if the noise is strongly non Poissonian, the pixels are not optimally weighted, and the CVT step appears to introduces significant gradients in the S/N. A similar alternative consists of using the /WVT keyword below.
/PLOT	Set this keyword to produce a plot of the two-dimensional bins and of the corresponding S/N at the end of the computation.
/QUIET	by default the program shows the progress while accreting pixels and then while iterating the CVT. Set this keyword to avoid printing progess results.
/WVT	When this keyword is set, the routine bin2d_cvt_equal_mass is modified as proposed by Diehl & Statler (2006, MNRAS, 368, 497). In this case the final step of the algorithm, after the bin-accretion stage, is not a modified Centroidal Voronoi Tessellation, but it uses a Weighted Voronoi Tessellation. This may be useful if the noise is strongly non Poissonian, the pixels are not optimally weighted, and the CVT step appears to introduces significant gradients in the S/N. A similar alternative consists of using the /NO_CVT keyword above. If you use the /WVT keyword you should also include a reference to 'the WVT modification proposed by Diehl & Statler (2006).'

/weight_for_sn	If set, the spectra in the same spatial bin will be weighted by S/N^2 before being added. This is equivalent to adopt the following transformation:
	`SIGNAL_NEW = (SIGNAL_OLD/NOISE_OLD)^2`
	`NOISE_NEW = SIGNAL_OLD/NOISE_OLD`

OUTPOUTS:	
BINNUMBER	[flt array]. Vector (same size as X) containing the bin number assigned to each input pixel. The index goes from zero to Nbins-1. This vector alone is enough to make *any* subsequent computation on the binned data. Everything else is optional!
XBIN	[flt array]. Vector (size Nbins) of the X coordinates of the bin generators. These generators uniquely define the Voronoi tessellation.
YBIN	[flt array]. Vector (size Nbins) of Y coordinates of the bin generators.
XBAR	[flt array]. Vector (size Nbins) of X coordinates of the bins luminosity weighted centroids. Useful for plotting interpolated data.
YBAR	[flt array]. Vector (size Nbins) of Y coordinates of the bins luminosity weighted centroids.
SN	[flt array]. Vector (size Nbins) with the final SN of each bin.
NPIXELS	[flt array]. Vector (size Nbins) with the number of pixels of each bin.

SCALE	[flt array]. Vector (size Nbins) with the scale length of the Weighted Voronoi Tessellation, when the /WVT keyword is set. In that case SCALE is *needed* together with the coordinates XBIN and YBIN of the generators, to compute the tessellation (but one can also simply use the BINNUMBER vector).

3.3.1 `mdap_sgandalf.pro`

This procedure is called by mdap_spectral_fit.pro and it fits an input galaxy spectrum with a set of stellar templates and Gaussian emission lines to get the absorption and emission line kinematics, the emission lines fluxes and equivalent widths, the weighs of the stellar templates, the best fit models for stars and gas, the best fitting stellar template and the best fitting stellar template convolved by the LOSVD, the residuals, and the reddening (if required).

The kinematics are recovered parametrically, using a Gauss function plus high order Gauss-Hermite moments for the stellar kinematics, and a Gaussian function for the emission line kinematics.

The code itself is an implementation of the ppxf code by M. Capellari (insert reference), and the spectroscopic decomposition code by L. Coccato (**?**).

The input galaxy spectrum is fitted according to the following steps:

1. an optimal stellar template is build as linear combination of stars in a library. Regularization over the stellar population properties can be used.

2. an optimal emission line spectrum is build as linear combination of N Gaussian emission lines. The number N of the emission lines and their rest-frame wavelength are read from an input file. This step should account for instrumenal LSF (to be developed).

3. the optimal stellar template is convolved with a Gauss-Hermite line of sight velocity distribution, parametrized by V, σ, h3, and h4.

4. the optimal emission line spectrum is convolved with a Gaussian line of sight velocity distribution, parametrized by V and σ.

5. the convolved optimal emission line spectrum and optimal template are multiplied by set of legendre polynomials, or by a reddening function, parametrized by $E(B-V)$ according to the mdap_dust_calzetti.pro function (Section **??**).

6. spectra from points 4 and 5 are added together and compared to the input galaxy spectrum.

7. points from 1 to 6 are repeated till minimum χ^2 is reached.

The list of input/output parameters is given in Table 3.3.

Table 3.3: Inputs and outputs parameters of mdap_sgandalf

INPUTS	

TEMPLATES1	[dbl array]. Vector containing the spectrum of a single template star or more ; commonly an array of dimensions TEMPLATES[nPixels,nTemplates] containing ; different templates to be optimized during the fit of the kinematics. ; nPixels has to be $\geq$ the number of galaxy pixels. Tips: • To apply linear regularization to the WEIGHTS via the keyword REGUL, TEMPLATES should be an array of two TEMPLATES[nPixels, nAge], three TEMPLATES[nPixels, nAge, nMetal] or four TEMPLATES[nPixels,nAge,nMetal,nAlpha] dimensions, depending on the number of population variables one wants to study. This can be useful to try to attach a physical meaning to the output WEIGHTS, in term of the galaxy star formation history and chmemical composition distribution. In that case the templates may represent single stellar population SSP models and should be arranged in sequence of increasing age, metallicity or alpha along the second, third or fourth dimension of the array respectively. • TEMPLATES and GALAXY do not need to span the same wavelength range. However an error will be returned by SGANDALF, if the velocity shift in pixels, required to match the galaxy with the templates, becomes larger than nPixels. In that case one has to truncate either the galaxy or the templates to make the two rest-frame spectral ranges more similar.
TEMPLATES2:	[N x 2 dbl array]. It contains: • TEMPLATES2[*,0] the values of the wavelenghts of the N emission lines to fit, in logarithmic units (ln or Log10 as in the input galaxy spectrum). • TEMPLATES2[*,1] The N signs of the gas lines to fit, +1 for emission lines, -1 for absorption lines (e.g. NaI.)
GALAXY:	[dbl array]. It contains the spectrum of the galaxy to be measured. The star and the galaxy spectra have to be logarithmically rebinned but the continuum does *not* have to be subtracted. The rebinning may be performed with the mdap_do_ log_rebin.pro (Section 3.2). For high redshift galaxies, one should bring the spectra close to the restframe wavelength, before doing the SGANDALF fit, to prevent too large velocity shifts of the templates. This can be done by dividing the observed wavelenghts by (1+z), where z is a rough estimate of the galaxy redshift, before the logarithmic rebinning. TO BE IMPLEMENTED IN THE PIPELINE.

NOISE:	[dbl array vector]. It contains the 1*sigma error (per pixel) in the galaxy spectrum. IMPORTANT: the penalty term of the sgandalf method is based on the *relative* change of the fit residuals. For this reason the penalty will work as expected even if no reliable estimate of the NOISE is available (see Cappellari & Emsellem [2004] for details). If no reliable noise is available this keyword can just be set to: NOISE = galaxy*0+1 ; Same weight for all pixels.
VELSCALE:	[float]. Velocity scale of the spectra in km/s per pixel. It has to be the same for both the galaxy and the template spectra.
START:	[6 elements vector]. [velStart_ stars,sigmaStart_ stars, h3, h4, velStart_ gas, sigmaStart_ gas] with the initial estimate for the velocity and the velocity dispersion in km/s.
OPTIONAL INPUTS	
BIAS:	[float]. This parameter biases the (h3,h4,...) measurements towards zero (Gaussian LOSVD) unless their inclusion significantly decreses the error in the fit. Set this to BIAS=0.0 not to bias the fit (Default value used in the DAP): the solution (including [V,σ]) will be noisier in that case. The default BIAS should provide acceptable results in most cases, but it would be safe to test it with Monte Carlo simulations. This keyword precisely corresponds to the parameter \lambda in the Cappellari & Emsellem (2004) paper. Note that the penalty depends on the *relative* change of the fit residuals, so it is insensitive to proper scaling of the NOISE vector. A nonzero BIAS can be safely used even without a reliable NOISE spectrum, or with equal weighting for all pixels.
DEGREE:	[integer]. degree of the *additive* Legendre polynomial used to correct the template continuum shape during the fit. Default: DEGREE = -1, i.e. no additive polynomial are fitted.
GOODPIXELS:	[integer array]. It contains the indices of the good pixels in the GALAXY spectrum (in increasing order). Only these pixels are included in the fit. If the /CLEAN keyword is set, in output this vector will be updated to contain the indices of the pixels that were actually used in the fit.
LAMBDA:	[dbl array]. When the keyword REDDENING is used, the user has to pass in this keyword a vector with the same dimensions of GALAXY, giving the restframe wavelength in Angstrom of every pixel in the input galaxy spectrum, i.e. lambda = EXP(logLam).
MDEGREE:	[integer]. degree of the *multiplicative* Legendre polynomial (with mean of 1) used to correct the continuum shape during the fit (default: 0). The zero degree multiplicative polynomial is always included in the fit as it corresponds to the weights assigned to the templates. Note that the computation time is longer with multiplicative polynomials than with the same number of additive polynomials. IMPORTANT: Multiplicative polynomials cannot be used when the REDDENING keyword is set.

MOMENTS:	[integer]. Order of the Gauss-Hermite moments to fit. Set this keyword to 4 to fit [h3,h4] and to 6 to fit [h3,h4,h5,h6]. Note that in all cases the G-H moments are fitted (nonlinearly) *together* with [V,sigma]. If MOMENTS=2 or MOMENTS is not set then only [V,sigma] are fitted and the other parameters are returned as zero. If MOMENTS=0 then only the templates and the continuum additive polynomials are fitted and the WEIGHTS are returned in output.
REDDENING:	[Float]. Set this keyword to an initail estimate of the reddening $E(B-V) >= 0$ to fit a positive reddening together with the kinematics and the templates. After the fit the input estimate is replaced with the best fitting $E(B-V)$ value. The fit assumes the exctinction curve of Calzetti et al. (2000, ApJ, 533, 682) but any other prescriptions could be trivially implemented by modifying the function SGANDALF_REDDENING_CURVE within the procedure. IMPORTANT: The MDEGREE keyword cannot be used when REDDENING is set.

REGUL:	[Float]. If this keyword is nonzero, the program applies second-degree linear regularization to the WEIGHTS during the SGANDALF fit. Regularization is done in one, two or three dimensions depending on whether the array of TEMPLATES has two, three or four dimensions respectively. Large REGUL values correspond to smoother WEIGHTS output. The WEIGHTS tend to a linear trend for large REGUL. When this keyword is nonzero the solution will be a trade-off between smoothness of WEIGHTS and goodness of fit. The effect of the regularization scheme is to enforce the numerical second derivatives between neighbouring weights (in every dimension) to be equal to -w[j-1]+2*w[j]-w[j+1]=0 with an error Delta=1/REGUL. It may be helpful to define REGUL=1/Delta and view Delta as the regularization error. IMPORTANT: Delta needs to be of the same order of magnitude as the typical WEIGHTS to play an effect on the regularization. One way to achieve this is: • divide the TEMPLATES array by a scalar in such a way that the typical template has a median of one (e.g. TEMPLATES/=median(TEMPLATES)); • do the same for the input GALAXY spectrum (e.g. GALAXY/=median(GALAXY)). In this situation Delta and REGUL should be *roughly* of order unity. Here is a possible recipe for chosing the regularization parameter REGUL: • Perform an un-regularized fit (REGUL=0) and then rescale the input NOISE spectrum so that χ^2/DOF = χ^2/N_ELEMENTS(goodPixels) = 1. This is achieved by rescaling the input NOISE spectrum as NOISE = NOISE*sqrt(χ^2/DOF) = NOISE*sqrt(SOL[6]); • Increase REGUL and iteratively redo the sgandalf fit until the χ^2 increases from the unregularized χ^2 = N_ELEMENTS(goodPixels) value by Δchi^2 = sqrt(2*n_elements(goodPixels)). The derived regularization corresponds to the maximum one still consistent with the observations and the derived star formation history will be the smoothest (minimum curvature) that is still consistent with the observations. For a detailed explanation see Section 18.5 of Press et al. (1992, Numerical Recipes 2nd ed.) available here http://www.nrbook.com/a/bookfpdf.php. The adopted implementation corresponds to their equation (18.5.10).

VSYST:	[Float]. Difference in km/sec between the first pixel of the ln-wavelenght of the template stars and the ln-wavelenght of the galaxy, i.e. dv = (loglam_templates[0]-loglam_gal[0])*c (it is computed in mdap_spectral_fitting.pro.
OPTIONAL KEYWORDS	
/CLEAN	set this keyword to use the iterative sigma clipping method described in Section 2.1 of Cappellari et al. (2002, ApJ, 578, 787). This is useful to remove from the fit unmasked bad pixels, residual gas emissions or cosmic rays. IMPORTANT: This is recommended *only* if a reliable estimate of the NOISE spectrum is available. See also note below for SOL.
/QUIET	set this keyword to suppress verbose output of the best fitting parameters at the end of the fit.
OUTPUTS	
SOL:	[9+MDEGREE elements vector]. It contains the values of [Vel_star, Sigma_Star, h3, h4, h5, h6,χ^2/DOF, Vel_gas, Sigma_gas] of the best fitting solution, where DOF is the number of Degrees of Freedom (number of fitted spectral pixels). Vel is the velocity, Sigma is the velocity dispersion, h3-h6 are the Gauss-Hermite coefficients. The model parameter are fitted simultaneously. The following safety limits on the fitting parameters are hard-coded(check!!): 1. Vel is constrained to be +/-2000 km/s from the first input guess 2. velScale/10 ¡ Sigma ¡ 1000 km/s 3. -0.3 ¡ [h3,h4,...] ¡ 0.3 (limits are extreme value for real galaxies) IMPORTANT: if χ^2/DOF is not $\sim$1 it means that the errors are not properly estimated, or that the template is bad and it is *not* safe to set the /CLEAN keyword. When MDEGREE ¿ 1 then SOL contains in output the 9+MDEGREE elements [Vel_star, Sigma_Star, h3, h4, h5, h6, χ^2/DOF,Vel_gas,Sigma_gas, cx1, cx2, ..., cxn], where cx1, cx2,...,c xn are the coefficients of the multiplicative Legendre polynomials of order 1, 2, ..., n. The polynomial can be explicitly evaluated as: x = range(-1d,1d,n_elements(galaxy)) mpoly = 1d ; Multiplicative polynomial for j=1,MDEGREE do mpoly += legendre(x,j)*sol[6+j] When the reddening correction is used, SOL contains 10 elements [Vel_star,Sigma_Star, h3, h4, h5, h6, χ^2/DOF, Vel_gas,Sigma_gas, ebv], where ebv is the best fitting reddening value.

gas_intens	[dbl array]. It contains the intensities of the emission lines (corrected for reddening if the reddening is fitted).
gas_fluxes	[dbl array]. It contains the fluxes of the emission lines (corrected for reddening if the reddening is fitted).
gas_ew	[dbl array]. It contains the equivalent widths of the emission lines (corrected for reddening if the reddening is fitted).
gas_intens_ err	[dbl array]. It contains the errors associated to gas_intens.
gas_fluxes_ err	[dbl array]. It contains the errors associated to gas_fluxes.
gas_ew_err	[dbl array]. It contains the errors associated to gas_ew.
OPTIONAL OUTPUTS	
BESTFIT:	[dbl array]. A named variable to receive a vector with the best fitting model: this is a linear combination of the templates, multiplied by multiplicative pols (if any) or reddening corrected (if required), convolved with the best fitting LOSVD, with added polynomial continuum terms and the best fitting gas emission lines.
BF_COMP1	[dbl array]. A named variable to receive a vector with the best fitting model for the stellar component: this is a linear combination of the stellar templates, multiplied by multiplicative pols (if any) or reddening corrected (if required), convolved with the best fitting LOSVD. It does not contain additive pols.
BF_COMP2	[dbl array]. A named variable to receive a vector with the best fitting ionized gas kinematics, convolved with the gas LOSVD, and mutiplied by the reddening curve (if applicable).
OPT_TEMPL	[dbl array]. A named variable to reveive a vector containing the optimal template. This is the linear combination of the templates, multiplied by multiplicative pols (if any) or reddening corrected (if required), NOT CONVOLVED with the stellar LOSVD. It does not contain additive pols.
MPOLY	[dbl array]. A named variable to receive a vector with the multiplicative pol that has been multipliedto BF_COMP1 and BESTFIT.
ADDITIVE_POL	[dbl array]. A named variable to receive a vector with the additive pol that has been added to BESTFIT.
ERROR:	[dbl array]. A named variable that will contain a vector of *formal* errors (1*sigma) for the fitted parameters in the output vector SOL.

POLYWEIGHTS	[Flt array]. A named variable to receive the weights of the additive Legendre polynomials. The best fitting additive polynomial can be explicitly evaluated as x = range(-1d,1d,n_elements(galaxy)) apoly = 0d ; Additive polynomial for j=0,DEGREE do apoly += legendre(x,j)*polyWeights[j] When doing a two-sided fitting (see help for GALAXY parameter), the additive polynomials are allowed to be different for the left and right spectrum. In that case the output weights of the additive polynomials alternate between the first (left) spectrum and the second (right) spectrum.
WEIGHTS	[float array] a named variable to receive the value of the weights by which each stellar template and ionized gas template was multiplied to best fit the galaxy spectrum. Stellar weights are WEIGHTS[0:Ntemplates1-1] Gas emission lines intensities are WEIGHTS[Ntemplates1 : *] N.B. Gas intensities are dereddened (i.e. the intensity in the input spectrum is lower than WEIGHTS[Ntemplates1 : *], because it account for the reddening at that wavelength. If reddening is not fitted, then intensities and weights are the same.

3.3.2 mdap_measure_indices.pro

This module is responsible for measuring the strength of the absoprtion line indices (performed with mdap_do_measure_indices.pro, see Section 3.4.1), their errors, and correct them for galaxy intrinsic broadening. Broadening correction is applied to the input galaxy spectra to match the spectral resolution of the spectral indices system (e.g. Lick). Input galaxy spectra must have emission lines removed.

Broadening correction is done according to the following formula

$$I_{\rm corr} = I_{\rm gal} \frac{I_{\rm templ}}{I_{\rm templ\ LOSVD}}$$

for atomic indices, and

$$I_{\rm corr} = I_{\rm gal} + I_{\rm templ} - I_{\rm templ\ LOSVD}$$

for molecular indices. $I_{\rm corr}$ is the index line strength corrected for intrinsic broadening, $I_{\rm templ}$ is the index line strength measured on the best fitting stellar template, and $I_{\rm templ\ LOSVD}$ is the index line strength measured on the best fitting stellar template convolved by the best fitting LOSVD.

The list of input/output parameters for this module is given in Table 3.4.

Table 3.4: Inputs and outputs parameters of mdap_measure_indices.pro

INPUTS	

wavelength	[N dblarray]. Vector containing the wavelenghts of the input spectra. The dispersion can be also not constant.
spectra	[T x N dblarray]. Vector containing the T input galaxy spextra, with emission line removed. Spectra are defined over the vector wavelength.
best_template	[T x N dblarray]. Vector containing the T best fitting stellar templates obtained when fitting the kinematics of the input spectra. Spectra are defined over the vector wavelength.
best_template_ LOSVD,	[T x N dblarray]. Vector containing the T best fitting stellar templates, convolved with the best-fitting stellar LOSVD, obtained when fitting the kinematics of the input spectra. Spectra are defined over the vector wavelength.
stellar_velocity	[T flt array]. Vector containing the best fitting stellar velocity for the T input spectra in km/sec. P.S. Set it to zero if the input spectra are in rest-frame
residuals	[T x N dblarray]. Vector containing the residuals from the best fit model to the input galaxy spectra. Residuals are defined over the vector wavelength.
fwhm_diff_ indices_	[N dblarray]. Vector that specifies the FWHM(λ) (in Å) that should be used to broaden the spectra, best_template, and best_template_LOSVD to match the spectral resolution of the spectral indices system.
OPTIONAL INPUTS	
dir=dir	Directory where to store the ps files showing the measurements
OUTPUTS	
abs_line_ indices	[T x 37 dbl array]. Absorption line indices of the T inut spectra, corrected for intrinsic broadening. The 37 measure indices are defined in aborption_line_ indices_definition, which is an to in the DAP.
abs_line_ indices_errors	[T x 37 dbl array]. Errors associated to abs_line_ indices.
abs_line_ indices_template	Absorption line indices measured on best_template
abs_line_ indices_template_ losvd	Absorption line indices measured on best_template_ LOSVD.
OPTIONAL OUTPUTS	
version	string specifying the module version. If requested, the module is not execute and only version flag is returned.

3.3.3 mdap_do_measure_indices.pro

This procedure measures the line strenght of a give index, according to the following definitions (from **?**):

$$\begin{aligned}
F(\lambda;\lambda_1,\lambda_2) &= \int_{\lambda_2}^{\lambda_1} F(\lambda)d\lambda/(\lambda_2-\lambda_1) \\
F_{PB} &= F(\lambda;\lambda_{\text{BLUE CONT 1}},\lambda_{\text{BLUE CONT 2}}) \\
F_{PR} &= F(\lambda;\lambda_{\text{RED CONT 1}},\lambda_{\text{RED CONT 2}}) \\
F_I &= F(\lambda;\lambda_{\text{INDEX 1}},\lambda_{\text{INDEX 2}}) \\
\lambda_{\text{BLUE C}} &= 0.5\cdot(\lambda_{\text{BLUE CONT 1}}+\lambda_{\text{BLUE CONT 2}}) \\
\lambda_{\text{RED C}} &= 0.5\cdot(\lambda_{\text{RED CONT 1}}+\lambda_{\text{RED CONT 2}}) \\
\lambda_{\text{INDEX C}} &= 0.5\cdot(\lambda_{\text{INDEX CONT 1}}+\lambda_{\text{INDEX CONT 2}}) \\
F_P(\lambda) &= (\lambda_{\text{RED C}}-\lambda_{\text{BLUE C}})\frac{\lambda-\lambda_{\text{BLUE C}}}{\lambda_{\text{RED C}}-\lambda_{\text{BLUE C}}}+\lambda_{\text{BLUE C}}
\end{aligned} \tag{3.1}$$

The Equivalent width (in Å) for atomic indices is:

$$EW = \int_{\lambda_{\text{INDEX 1}}}^{\lambda_{\text{INDEX 2}}} \left(1-\frac{F_I}{F_P}\right) d\lambda$$

The Equivalent width (in magnitudes) for molecular indices is:

$$EW = -2.5\ln\left[\left(\frac{1}{\lambda_{\text{INDEX 2}}-\lambda_{\text{INDEX 1}}}\right)\int_{\lambda_{\text{INDEX 1}}}^{\lambda_{\text{INDEX 2}}}(F_I/F_P)\,d\lambda\right]$$

The list of input/output parameters for this module is given in Table 3.5.

Errors on the indices are calculated using the Empirical formula by Cardiel et al. (1998), A&AS, 127, 597 Equations 41 -46.

Table 3.5: Inputs and outputs parameters of mdap_do_measure_indices.pro

INPUTS	
spc	[dbl array]. Vector containing spectra to calculate the line strenght index. It should be without emission lines, and at the same spectral resolution of the desired spectroscopic system.
lambda	[dbl array]. Vector (same number of elements of spc) containing the wavelengths (in Å). The vector must have constant Å/pixel step.
passband	[flt array]. 2 elements array defining the index passband boundaries.
blue_cont	[flt array]. 2 elements array defining the blue pseudocontinua boundaries.
red_cont	flt [array]. 2 elements array defining the red pseudocontinua boundaries.

OPTIONAL INPUTS	
norm	[float]. Value of λ (in Å) at which compute the normalization. The input spectrum is normalized by spc(norm). Defaul: no normalization.
title	[string]. Title to write into the plot produced as output.
plbound	[array]. Two elements array specifying the boundaries of the plot (Y axis), which will be set to [plbound[0]*midplot,plbound[1]*midplot] where midplot is the value of the spectrum at the wavelenght middle range. Default plbound=[0.6,1.35]; midplot=1.
rebin	[float]. If set, the input spectrum will be rebinned according to a new step (Å/pixel, defined by rebin). The starting point of lambda will remains unchaged. The input "lambda" and "spc" parameters are not overwritten. Default: no rebinning.
noise	[float]. It is useful only when errors need to be retrieved. Default: noise = sqrt(spc).
OPTIONAL KEYWORDS	
/noplot	If set, all the plotting commands (plot, oplot, plots and xyouts) in the routine are not executed.
OUTPUTS	
ew	[float]. Line equivalent width in angstrom.
index_mag	[float]. Linestrenght index value in magnitudes.
OPTIONAL OUTPUTS	
errors	[float]. This variable will contain the errors on the indices computed using Cardiel et al. 1998, A&AS, 127, 597.

3.4 mdap_gandalf.pro

T.B.D.

3.4.1 mdap_dust_calzetti.pro

This procedure uses the dust model of Calzetti et al. (2000, ApJ, 533, 682), and for a given E(B-V) value returns the flux attenuation array, which can be used to get reddened templates. Here the spectra are assumed to be binned on a ln-rebinned wavelentgh grid as defined by input parameters. The input receiding velocity is used to derive the dust reddening in the galaxy rest-frame.

Can be used also to de-reddened the galaxy spectra by the Milky-Way dust extinction, using as E(B-V) the opposite of the Schlegel et al. values found in NED and vstar = 0.

Initial version kindly provided by S. Kaviray, Oxford, 2006.

The list of input parameters and the return value for this function is given in Table 3.6.

Table 3.6: Inputs parameters and return value of mdap_dust_calzetti.pro function

INPUTS	
l0_gal	[double]. Starting ln-wavelength where the galaxy spectrum is defined. Default $\ln(\lambda)$, if \log10 keyword is set, $\log_{10}(\lambda)$ is assumed.
lstep_gal	[double]. Constant logarithmic step (natural log, unless \log10 keyword is set).
npix	[integer]. Number of pixels of the galaxy spectrum.
ebv	[double]. E(B-V) reddening coefficient.
vstar	[double]. Vector of lambda will be de-redshifted by this amount to provide correction at restframe.
OPTIONAL KEYWORDS	
\	If set, input quantities l0_gal and lstep_gal are assumed to be in $\log_{10}$. Default is natural log.
RETURN VALUE	[dbl array]. array with npix elements containing the reddening vector. This is used to correct an input galaxy spectrum, like spc_obs = spc_corr *times* mdap_dust_calzetti(l0_gal, lstep_gal, npix, ebv, vstar)

Chapter 4

Utility modules

4.1 mdap_convol_sigma.pro

T.B.D.

4.2 mdap_calibrate_sn

T.B.D.

4.3 mdap_interpolate_2dmaps.pro

This routine:

1. uses the GRID_TPS IDL function to interpolate a set of N input values, defined over an irregular grid, on to a regular N'xM' grid.
2. interpolates the values interpolated over the regular grid (point i) and re-interpolates them over an irregular grid using the IDL BILINEAR function.

The table 1.2.1 lists the inputs and outputs parameters of mdap_interpolate_2dmaps.pro.

Table 4.1: Inputs and outputs parameters of mdap_interpolate_2dmaps.pro

INPUTS	
input	N elements array containing the values to be interpolated.
x	N elements array containing the X coordinates where the N input values are defined.
y	N elements array containing the Y coordinates where the N input values are defined.
x2d_full	N'xM' elements array containing the X coordiantes of a regularly spaced grid.
y2d_full	N'xM' elements array containing the Y coordiantes of a regularly spaced grid.
x_out	M elements array containing the X coordinates where the input values need to be interpolated.
y_out	M elements array containing the X coordinates where the input values need to be interpolated.
OUTPUTS	
output	M elements array containing the interpolated input values on the x_out,y_out coordiantes.
OPTIONAL OUTPUT	
full_grid	N'xM' elements array containing the interpolated input values on the regular grid with coordinates x2d_full, y2d_full

4.4 Visualization and Web Interface. T.B.D.

We want at least the beginning of a plan here for what we have in mind for visualization tools and the web interface. This is meant to spark discussion at the Portsmouth meeting, get us thinking ahead of time, and also get some feedback from the panelists.

4.4.1 `mdap_get_2d_map.pro`

This module is responsable to arrange the outputs of mdap_spectral_fitting.pro into a two-dimensional map, given some spatial binning informations.

Table 4.2 lists the inputs and outputs required for this module.

Table 4.2: Inputs and outputs parameters of mdap_get_2d_map.pro

INPUTS	
values	[Nbins elements dbl array]. Measured quantity for each spatial bin (first elements is associated with bin 0 , second element with bin 1, and soforth). It can be either an output of block 4 or block 5.
binning	[N x M dbl array]. Two dimensional map showing the binning sheme. Pixels beloning to the i+1-th bin have value i (i= 0, 1, ..., Nbins-1). Pixels associated to no spatial bin have value -1 (produced in block 2 by mdap_spatial_binning.pro). Nbins is the number of spatial bins.
x2d	[N x M flt array]. X coordinates of the field of view, in arcsec (produced in block 1 by mdap_read_datacube.pro). Coordinate 0 is the center of the field of view.
y2d	[N x M flt array]. Y coordinates of the field of view, in arcsec (produced in block 1 by mdap_read_datacube.pro). Coordinate 0 is the center of the field of view.
OUTPOUTS	
map	[N x M flt array]. Measured values over the two-dimensional field of view. Pixels belonging to the same spatial bin have the same value (i.e. no spatial smoothing or interpolation are performed).

4.4.2 `mdap_display_2dmap_ps.pro`

T.B.D.