The MANGA Data Analysis Pipeline: a prototype

L. Coccato & MANGA team

May 7, 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.
- Radial profiles of λ, V/σ, and σ.
- 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).

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	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 mdap_gandalf mdap_ppxf `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 mdap_do_measure_index	mdap_round_str.pro mdap_convol_sigma
Block 6	mdap_spatial_radial_binning mdap_spectral_fitting mdap_measure_indices mdap_do_kinemetry.pro	mdap_calculate_spectrum_sn mdap_do_measure_index mdap_dust_calzetti `mpfit package` mdap_sgandalf mdap_gandalf_wrapf mdap_gandalf mdap_ppxf mdap_bvls(_external) mdap_kinemetry.tex mdap_do_k_rprofiles.pro	mdap_convol_sigma mdap_get_losvd mdap_range.pro mdap_stc.pro mdap_sgn.pro mdap_interpolate_2dmaps mdap_read_indices_definitions mdap_ppxf_convol_fft
Block 7			
Block 8			
Block 9			

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, main modules, and utilities.

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 2.5.1).
- 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 lines	Measurements performed on the third binning scheme: emission line properties (reddening, intensity, flux, and equivalent width).
8	binning 3 kinematics	Measurements performed on the third binning scheme: mean kinematics, and kinematics of individual emission lines.
9	binning map radial	Location and geometry of the radial binning scheme.
10	binning radial data	Measurements performed on the radial binning scheme (absorption line indices and stellar velocity dispersion).
11	Stars rotation	Kinemetric measurements on the stellar kinematics (kinematic position angle, rotation curve, inflows/outflows).
12	Gas rotation	Kinemetric measurements on the gas kinematics (kinematic position angle, rotation curve, inflows/outflows).
13	profiles	Stellar radial profiles of $\lambda(R)$, $V/\sigma(R)$, and $\sigma(R)$.

Ext	Name	Description

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 galaxy has coordinates (0,0).
- Column 2. Y. The Y coordinates (in arcsec) of the centers of the spatial bins. The center of the galaxy 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. Emission line properties

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 galaxy has coordinates (0,0).
- Column 2. Y. The Y coordinates (in arcsec) of the centers of the spatial bins. The center of the galaxy 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. E(B-V) color excess for the stellar component and its error
- Columns 8-9. E(B-V) color excess for the ionized gas stellar component and its error
- Column 10. CHI2. Chi-squared from the fit.
- Columns 11-end. Intensities, Intensities errors, 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). Warning: intensities and fluxes are per arcsec^2, i.e. the numbers measured in the spectrum are divided by the area of the spatial bin (column 3).

Extension 8: outputs related to the third binning scheme. Kinematics

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

- Column 1. X. As in extension 7.
- Column 2. Y. As in extension 7.
- Columns 3-4. MEAN_VEL and ERR_MEAN_VEL. Flux-weighted mean velocity of the emission lines and its error in km/sec.
- Columns 5-6. MEAN_DISP and ERR_MEAN_DISP. Flux-weighted mean velocity dispersion of the emission lines and its error in km/sec.
- Columns 7-end. Velocities, velocity errors, velocity dispersions and velocity dispersino errorrs measured for the individual emission lines.

Extension 10: 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).

Extension 11 (12): outputs related to the kinemetric analysis of stellar (gas) kinematics

The quantities measured and stored in extension 10 (11) are:

- Column 1. Elliptical semimajor axis (computed fixing the systemic velocity and the kinematic center, position angle, and axial ratio).
- Column 2. Mean kinematic position angle.
- Column 3. Standard deviation of the kinematic position angles at each semimajor axis.
- Column 4. Mean kinematic axial ratio.
- Column 5. Standard deviation of the kinematic axial ratios at each semimajor axis.
- Column 6. Systemic velocity.
- Column 7. Standard deviation of the systemic velocities at each semimajor axis.
- Column 8. Rotation velocity measured at each semimajor axis.
- Column 9. Error on the rotation velocity measured at each semimajor axis.
- Column 10. Expansion velocity measured at each semimajor axis.
- Column 11. Error on the expansion velocity measured at each semimajor axis.

See Section 2.6.2 for more information on how these quantities are computed.

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.
- low_radial_bins_user_inputs. String array. Values of the lower boundaries of the radial binning scheme. Default: 6 logarithmic binnings are created.
- upper_radial_bins_user_inputs. Values of the upper boundaries of the radial binning scheme. Default: 6 logarithmic binnings are created.
- add_default_bins. If set, it adds the default 6 logarithmic radial binnings in addition to those defined by the user.
- 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 3.6). 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.

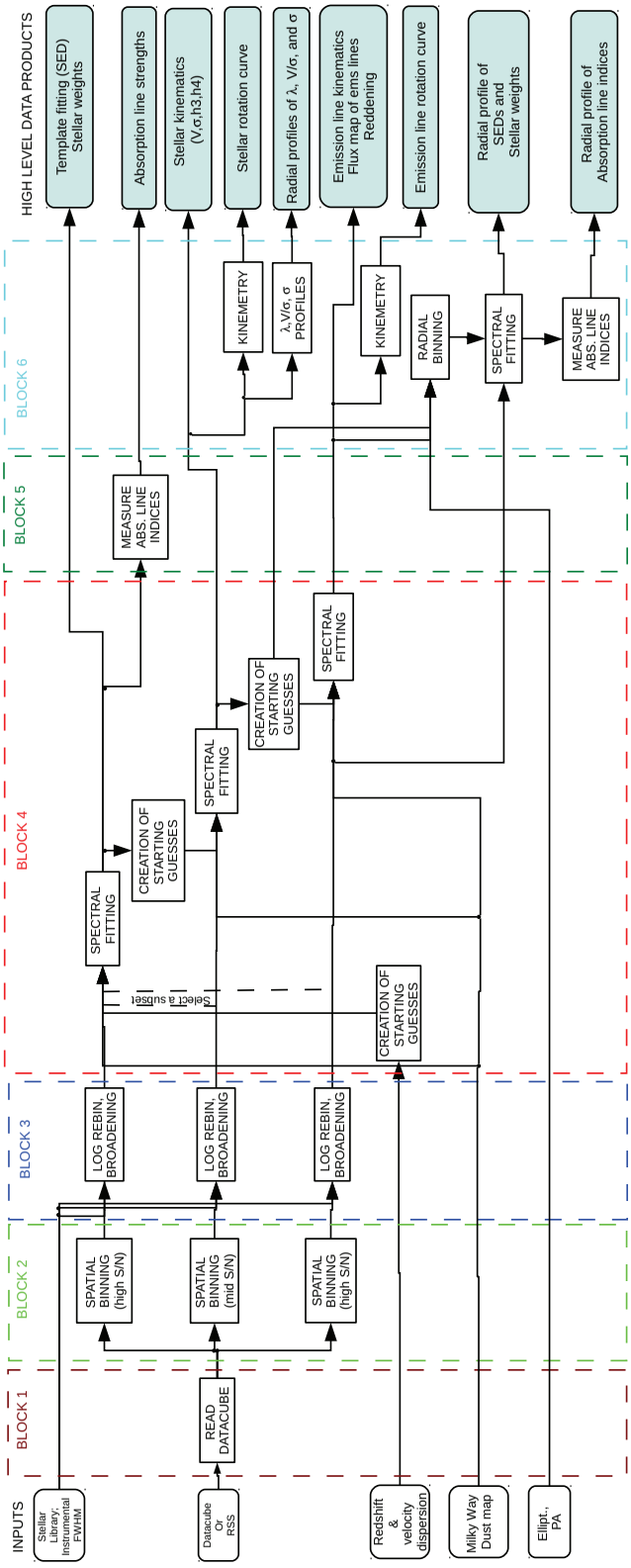

Figure 1.1: Workflow chart of the Data Analysis Pipeline (High Level Data products).

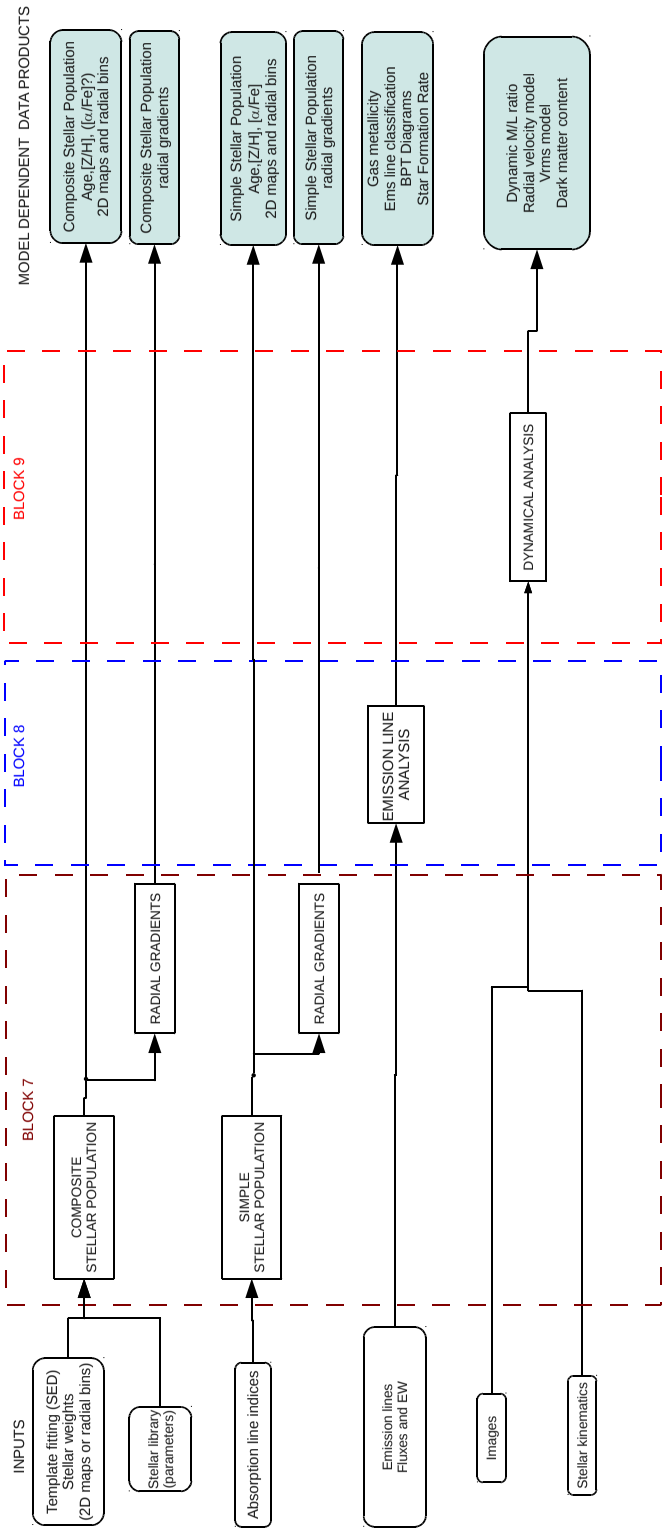

Figure 1.2: Workflow chart of the Data Analysis Pipeline (Model Dependent Data products).

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 origin of the reference coordiante system of the input datacube (or RSS files) is set to be the center of the field of view. The mdap_read_datacube.pro interface computes the center if the galaxy and set the reference coordinate system to the galaxy center. The galaxy center is computed via centroid fitting on the signal2d_reconstructed image via the gcntrd.pro IDL routine.

If the change of coordinates is larger than 3" in X or Y, the galaxy center is automatically set to be the same of the field of view. This is to avoid failures in the centroid fitting caused by the presence of foregroud stars in the field of view.

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.

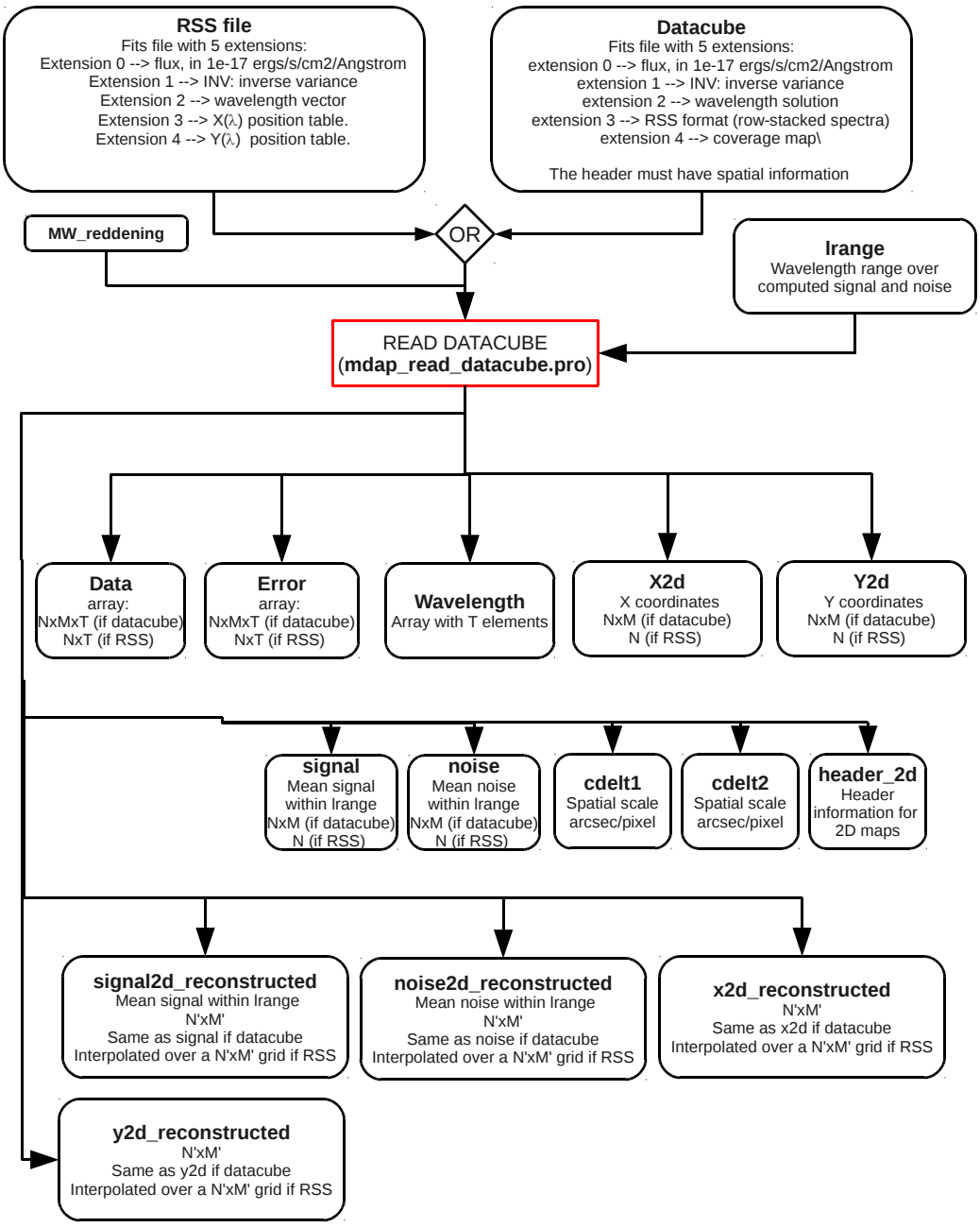

Figure 2.1: Workflow chart of the block 1 of the Data Analysis Pipeline. See Section 2.1.1 for a description of inputs and outputs.

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. It also computes the center if the galaxy and set the reference coordinate system to the galaxy center. The galaxy center is computed via centroid fitting on the signal2d_reconstructed image via the gcntrd.pro IDL routine.

The mdap_read_datacube.pro interface interpretes a shift larger than $3''$ of the X, Y coordinates of the galaxy center with respect the center of the field of view as bad fitting, caused by the presence of a foreground star or a non proper image reconstruction. Therefore, in this case, the coordinates of the galaxy center are set to be the same of the center of the field of view.

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 galaxy, determined via a centroid fitting). [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 galaxy, determined via a centroid fitting). [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.

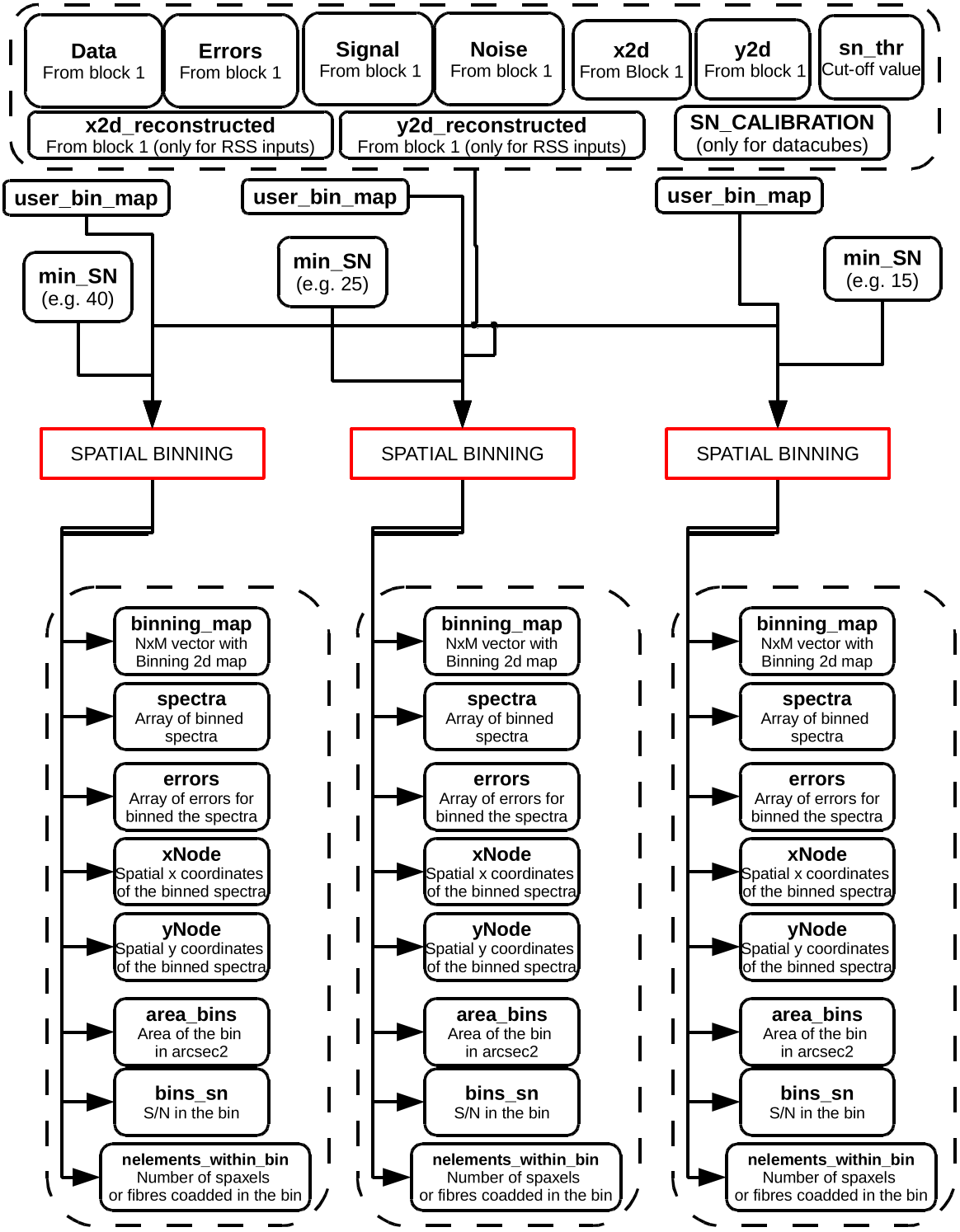

Figure 2.2: Workflow chart of the block 2 of the Data Analysis Pipeline.

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).

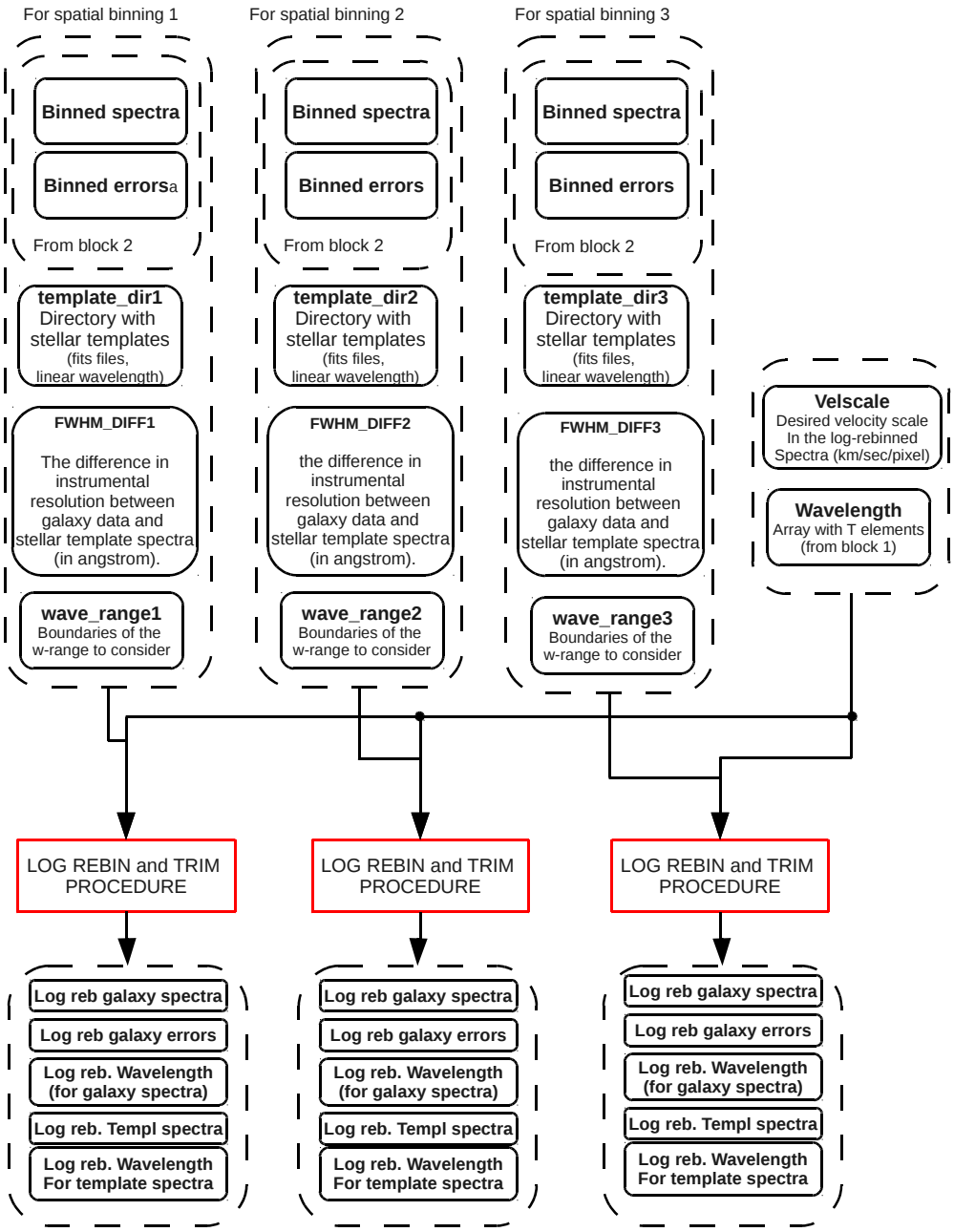

Figure 2.3: Workflow chart of the block 3 of the Data Analysis Pipeline.

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 3.4 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.

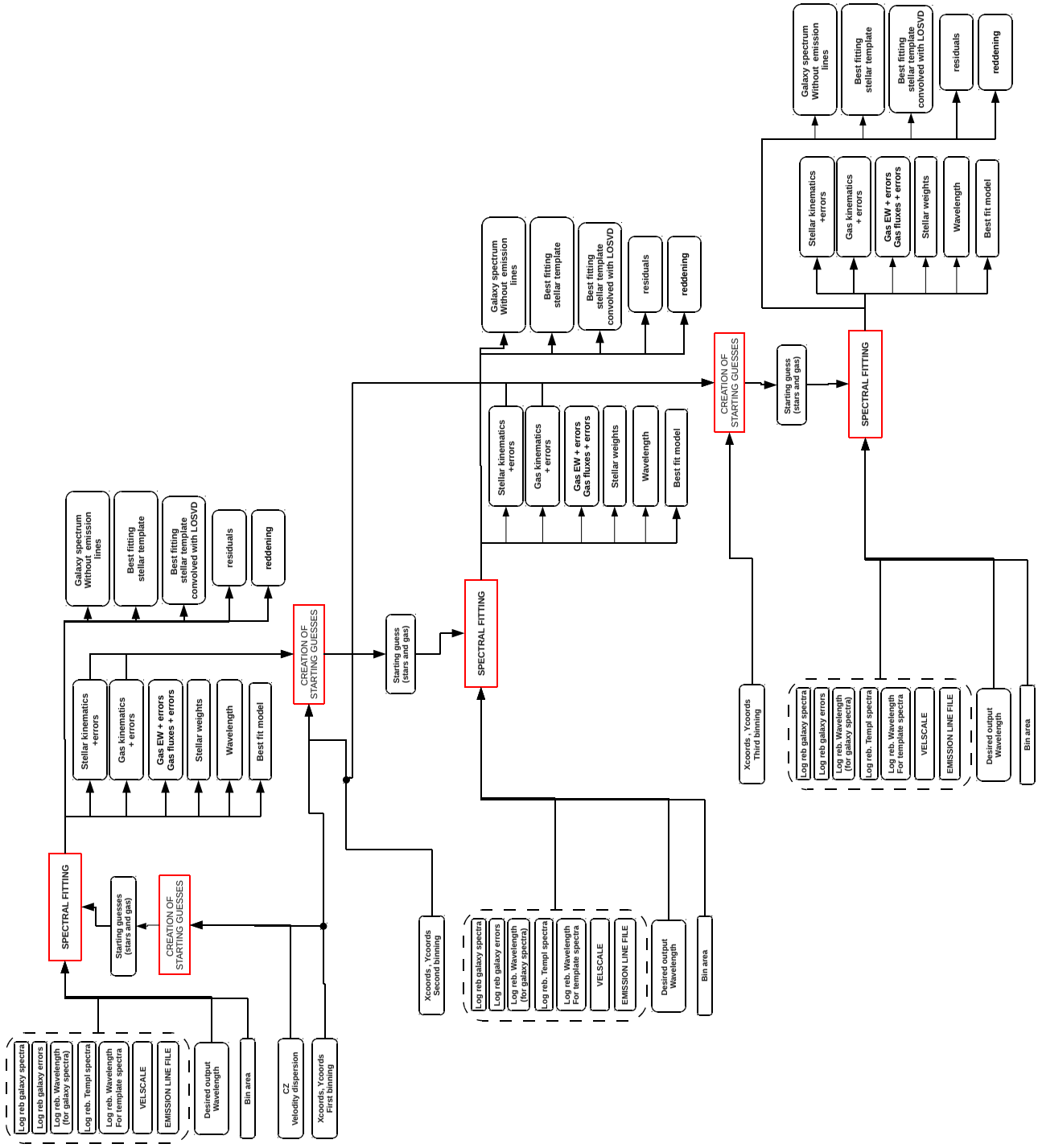

Figure 2.4: Workflow chart of the block 4 of the Data Analysis Pipeline.

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 format must be compatible with the module that performs the fit of the emission lines. If Gandalf or S-Gandalf are used, the input file must be an ascii file with the following 9 columns as in the following example (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
```

- `ID` Unique integer identifyer of the emission line.
- `CODE`. String. Name of the element. Special characters as "[", "." are not permitted.
- `wav`. Float. Rest frame wavelength of the emission line to fit. Warning, the name of the emission line in the final DAP output will be defined by the string `CODE` +"_"+ `ROUND(wav)`.
- `action`. String. Possible values are "f" (fit), "m" (mask), and "i" (ignore). All lines are masked in the pPXF run, and then fit in the Gandalf run, unless they are marked with "i", in this latter case they will be ignored.
- `line`. String Possible values are "l" (line) or "dN" (doublet). In this case, N indicates the line ID to which the doublet is linked, linked to the line with ID=N. For example, if emission line with `ID=4` has `line=''d3''`, then the emission line with `ID=3` must have `line = ``l''`.
- `int`. Float. Relative intensity of the gas emission (positive) or absorption (negative) lines. It is set to 1 for lines (`line = ``l''`). For doublets, ( `line = ``dN''`) it indicates the ration elements of the doublets.
- `vel`. Float. Guess of the velocity offset with respect the galaxy systemic velocity (ignored in the DAP)
- `sigma`. Float. Guess of the Velocity dispersion (ignored in the DAP)
- `mode`. String. Possible values are "f" (fit) or "tN" (tied). If a line has `mode = tN`, its kinematics is tied to the line with `ID=N`. In this case, the line with `ID=N` must have `mode = f`.

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_ intens	[N x T flt array]. It contains the intensities of the T fitted emission lines for each of the N input galaxy spectra. Values are corrected for reddening.
emission_line_ intens_err	[N x T flt array]. Errors associated to emission_line_intens
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_ 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	[N x 2 elements array]. Best fit values for the reddening of stars (reddening_ output[*,0]) and gas (reddening_ output[*,1]) for all the N galaxy spectra. If the reddening fit is not requred, the output value is set to 0. (reddening_output[*,0:1] = [0,0]). If only the reddening of the stars is fitted, the reddening of the gas is set to 0 (reddening_output[*,1] = 0).
reddening_ output_err	[N x 2 elements array]. Errors associated to reddening_ output. If not fitted, the error is automatically set to 99.
residuals	[N x QQ flt array]. It contains the difference between the observed galaxy spectra 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 block 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).

The interface module in block 5 is `mdap_measure_indices.pro` (Section 2.5.1). Figure 2.5 illustrates the flowchart of this module. Table 2.6 describes the inputs and outputs of this interface. The actual measurements are performed by the main module `mdap_do_measure_indices.pro` (Section 3.8).

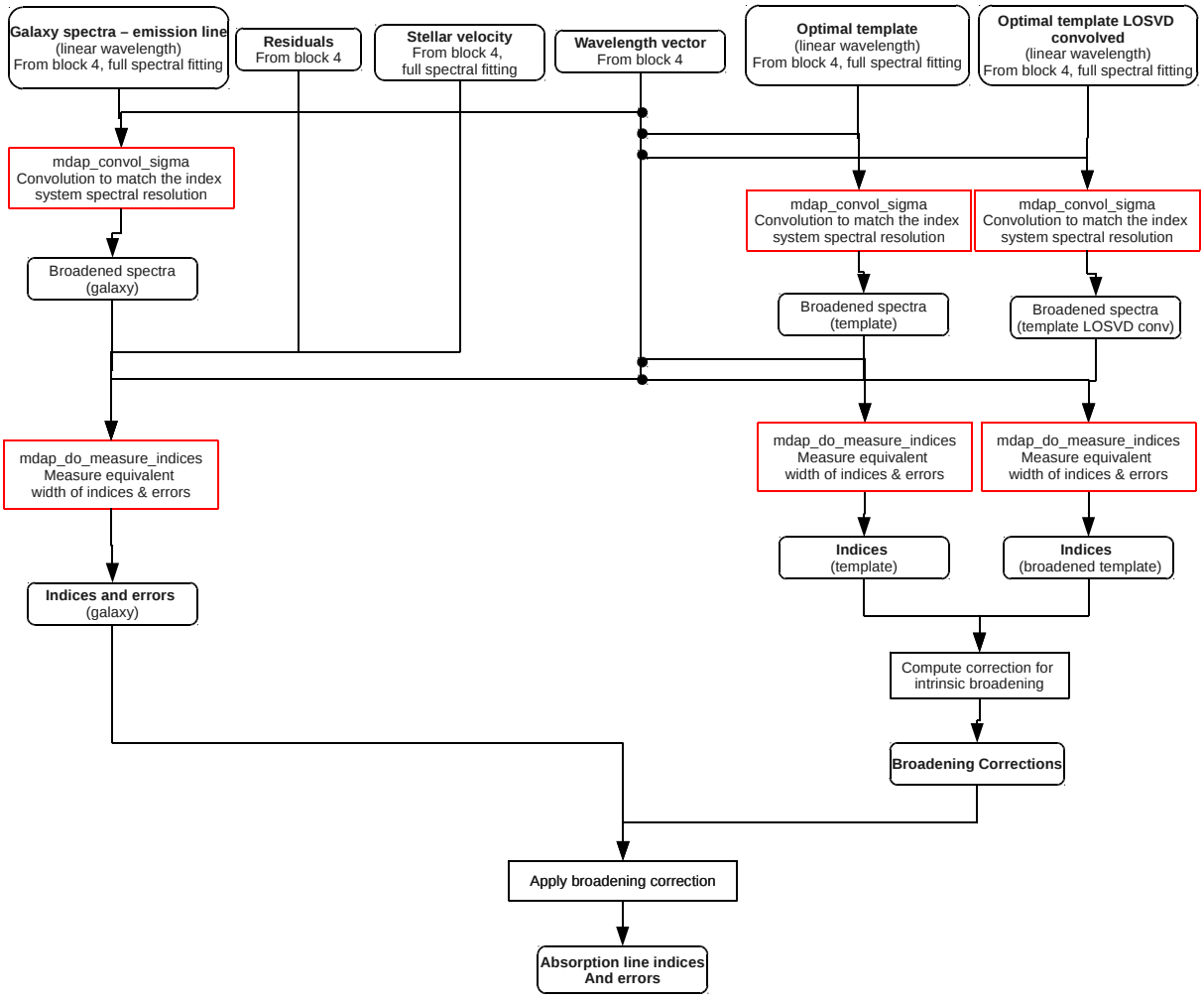

Figure 2.5: Workflow chart of mdap_measure_indices.pro, the interface module in block 5 of the Data Analysis Pipeline.

2.5.1 mdap_measure_indices.pro

This interface is responsible for measuring the strength of the absoprtion line indices, 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.

The measurement of the index is performed by the main module mdap_do_measure_indices.pro, see Section 3.8.

Warning: "D4000" and "TiO0p89" (i.e. TiO0.89) indices are defined as the ratio of the flux in the red and blue pseudocontinua. This alternative configuration is hardwired in the mdap_measure_indices.pro interface, and it is triggered by the index name (i.e. be sure that the user input file with indices definitions have these two names correcty spelled).

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 Convolution the input spectra to match a desired system is done by the utility module mdap_convol_sigma (Section 4.1).

The list of input/output parameters for this module is given in Table 2.6.

Table 2.6: Inputs and outputs parameters of mdap_measure_indices.pro

INPUTS	
wavelength	[N elements array]. Vector containing the wavelenghts of the input spectra. The dispersion can be also not constant.
spectra	[T x N elements array]. Vector containing the T input galaxy spextra, with emission line removed. Spectra are defined over the vector wavelength.
best_template	[T x N elements array]. 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 elements array]. 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 elements 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 elements array]. 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 elements array]. 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 Nind array]. Absorption line indices of the T inut spectra, corrected for intrinsic broadening. The Nind measure indices are defined in aborption_line_ indices_definition, specified in the configuration file (see Section 1.3).
abs_line_ indices_errors	[T x Nind 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.

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

This block measures the radial properties of the absorption line indices (part 1) and the kinematics i.e. rotation curves (parts 2 and 3).

The first part uses the restframed galaxy spectra (and errors) with removed emission lines produced in block 4 from the third spatial binning. Spectra are grouped according to elliptical bins (using the user input mean ellipticity an position angle) and added together. Then the spectra are fitted (`mdap_spectral_fitting`, Section 2.4.2) and the indices are measures (`mdap_measure_indices.pro`, Section 2.5.1). In the spectral fitting, emission lines are fitted again (except NaD absorptions), to remove additional residuals; each emission line has can have its own velocity and velocity dispersion. Stellar kinematics is bound to be within $-50 < V < 50$ km/sec. Reddening is not fitted.

Then, the output spectra (with residual emission lines removed, along with the best fitting stellar template (to correct for intrinsic broadening) are used to measure the absorption line indices.

The second part uses the stellar and emission line kinematics from block 4 to measure the kinematic position angle, kinematic center, kinematic axial ratio, rotation curve and amplitude of inflows/outflows. The interface that handles the second part is `mdap_do_kinemetry.pro` (Section 2.6.2), which implements kinemetry.pro by Kraijnovic et al. 2008.

The third part uses the stellar two-dimensional kinematics to compute the radial profile of the specific angular momentum ($\lambda(R)$), the $V/\sigma(R)$, and $\sigma(R)$ (see Section 2.6.3).

2.6.1 mdap_spatial_radial_binning.pro

T.B.D.

2.6.2 `mdap_do_kinemetry.pro`

This interface module is responsible to pass the measured stellar and emission line velocities to the kinemetry module, which measures the rotation curves, outflows and inflows motions, mean kinematic position angle, and mean kinematic axial ratio of stars and gas. The kinemetry module uses the kinemetry.pro module by Kraijnovic et al. 2005 (Section 3.9)

The following steps are executed:

- A first kinemetry run is executed, to get the the galaxy systemic velocity is kept constant at all radii (a first kinemetry.pro run is executed to measure the systemic velocities Vs_iat each i-th radius. The galaxy systemic velocity (Vsyst) and its 1sigma error (Vsyst_std) are computed by the median and the standard deviation of the systemic velocities Vs_i).
- A second kinemetry run is executed, fixing the kinematic center and the systemic velocity. In this run, the kinematic position angles (PA_i) and flattening (q_i) are measured for each radius. The galaxy mean kinematic position angle (PA_kin) and its error (PA_kin_std) are measured as the median and standard deviation of all the PA_i previously measured. The galaxy mean kinematic axial ratio (q_kin) and its error (q_kin_std) are measured as the median and standard deviation of all the q_i previously measured.
- A third and final kinemetry run is executed, fixing the kinematic center, position angle, and axial ratio determined in the previous runs. This last run determines the rotation velocity, and expansion velocity (i.e. inflows/outflows) for several radii.

Table 2.7 lists the inputs and outputs of the interface module mdap_do_kinemetry.pro.

Table 2.7: Inputs and outputs parameters of mdap_do_kinemetry.pro

INPUTS	
image	NxM array. Galaxy image. It is used to determine the location of the center.
x2d	NxM array. X coordinates corresponding to image (0,0 is the center of the field of view.
y2d	NxM array. Y coordinates corresponding to image (0,0 is the center of the field of view.
x	T elements array. X coordinates at which the velocites are measured.
y	T elements array. X coordinates at which the velocites are measured.
velocity	T elements array. The measured velocities (in km/sec).
velocity_err	T elements array. The measured velocity errors (in km/sec).
OUTPUTS	
PA_kin	Median kinematic position angle used to determine Vrot and Vexp (third kinemetry.pro run).
PA_kin_std	Standard deviation of the kinematic position angles measured in the second kinemetry run.

q_kin	Median kinematic axial ratio used to determine Vrot and Vexp (third kinemetry.pro run).
q_kin_std	Standard deviation of the kinematic axial ratio measured in the second kinemetry run.
Vsyst	Systemic velocity used to determine Vrot and Vexp (third kinemetry.pro run)
Vsyst_std	tandard deviation of the systemic velocities determined in the first kinemetry run.
Rad_kin	W elements array. Semi major axis of the ellipses where Vrot and Vexp are measured.
Vrot	W elements array. Rotational velocity measured at Rad_kin.
Vrot_err	W elements array. Errors on Vrot.
Vexp	W elements array. Outflow/Inflow velocity measured at Rad_kin.
Vexp_err	W elements array. Errors on Vexp.
OPTIONAL OUTPUTS	
version	String containin the module version. If requested, the module is not executed and only version flag is returned.

2.6.3 mdap_do_k_rprofiles.pro

).

T.B.D.

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.

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∼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).

3.4 mdap_gandalf_wrap.pro

This main module fits an input galaxy spectrum with a series of stellar templates and gas emission lines to derive the stellar and emission lines kinematics, and the fluxes and equivalent width of the emission lines.

The fit of the stellar kinematics is done using an implementation of the pPXF routine (Cappellari & Emsellem 2004, see Section 3.5), the fit of the emission line kinematics, the weights of the stellar templates, (and reddening, if reuired) is done using an implementation of the Gandalf routine (Sarzi et al. 2006, see Section 3.6).

The reddening of stars and gas (Balmer decrement) are fitted, if required, using the Calzetti extinction law (Calzetti et al. 2000, see Section 3.7).

The steps performed by the `mdap_gandalf_wrap.pro` main module are:

- selection of regions to mask. These are:
  - the regions around emission lines; their location is computed using the velocity initial guess, and the width of the region to mask is set to 250 km/sec.
  - The regions defined by the mask_range input keyword, if provided.
- Run pPXF to measure the stellar kinematics. Additive and multiplicative polynomials are used in the fit, if specified in the user in the configuration file. The fit of the reddening is not performed. Note: The stellar kinematics is free to vary within the boundaries specified by the user (keywords range_v_star and range_s_star). This is an implementation to the original pPPXF procedure by Cappellari & Emsellem (2004).
- Remove the masks around the emission lines, but keeping the masks defined by the mask_range input keyword (if defined).
- Run Gandalf to measure the gas emission lines kinematics, and intensities. Additive polynomials are not used. Multiplicative polynomials are used (as in the previous pPXF run) only if the reddening is not fitted, otherwhise they are set to 0. Note: The gas kinematics is free to vary within the boundaries specified by the user (keywords range_v_gas and range_s_gas). Also, the instrumental dispersion of the gas is assumed to vary with wavelength, therefore each emission line has its own value for the instrumental dispersion. These are two implementations to the original Gandalf procedure by Sarzi et al. (2006).
- Compute gas fluxes and equivalent widths (values are corrected for reddening, if reddening fit is required) from the emission line intensities.
- Computation of the mean kinematics of the emission lines. The mean gas velocity and velocity dispersions are defined as the flux weighted average of the velocities and velocity dispersions of the detected individual emission lines.

Table 3.3 lists the inputs and outputs parameters of the mdap_gandalf_wrap.pro module.

Table 3.3: Inputs and outputs parameters of mdap_gandalf_wrap.pro

INPUTS	
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.
galaxy	[N elements array]. Galaxy spectrum, logarithically rebinned, to be fitted.
loglam_gal	[N elements array]. $log(\lambda)$ values where the galaxy spectrum is defined.
noise	[N elements array]. Error vector associated to galaxy, defined over the loglam_gal vector.
velscale	[float]. Defines the (uniform) sampling of the input spectra, in km s^{-1}/pixel.
start_	6 elements array containing the starting guesses: start_[0] stellar veocity (km/sec) start_[1] stellar velocity dispersion (km/sec) start_[2] stellar h3 Gauss Hermite moment start_[3] stellar h4 Gauss Hermite moment start_[4] gas velocity (km/sec) start_[5] gas velocity dispersion (km/sec).
OPTIONAL INPUTS	
EMISSION_SETUP_FILE	As in Table 2.5.
BIAS	As used Table **??**.
MDEGREE	Integer. Degree of multiplicative polynomials to be used in the pPXF fit and in the Gandalf fit (if reddening is not fitted). Default: 0 (no multiplicative polynomials are used).
DEGREE	Integer. Degree of multiplicative polynomials to be used in the pPXF only. Default: -1 (no additive polynomials are used).
reddening	1 or 2 elements array. If specified in input, it triggers the fittind of the stellar reddening (1 element array) and the gas reddening (balmer decrement) (2 elements array). In output it will contain the best fit reddening values.
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}.
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) $< \lambda_{(2n)}$.
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.
INT_DISP	N elements array, containing the instrumental velocity dispersion (in km/sec) for all the N emission lines measured at their observed wavelengths (defined by the gas starting velocity guess).
KEYWORDS	
/FOR_ERRORS	If specified, it will trigger the computation of the emission lines error (see Section 3.6). Mandatory for the DAP workflow.
/fix_star_ kin	If set, the stellar kinematics will be fixed to the starting guesses values.
/fix_gas_ kin	If set, the gas kinematics will be fixed to the starting guesses values.
OUTPUTS	
sol	9 elements array containing the best fit kinematic parameters. sol[0]:stellar velocity (km/sec). sol[1]:stellar velocity dispersion (km/sec). sol[2]:stellar h3 Gauss-Hermite moment. sol[3]:stellar h4 Gauss-Hermite moment. sol[4]:stellar h5 Gauss-Hermite moment (not used). sol[5]:stellar h6 Gauss-Hermite moment (not used). sol[6]:χ^2. sol[7]: mean flux weighted velocity of the emission lines (km/sec). sol[8]: mean flux weighted velocity dispersion (km/sec).
gas_intens	N elements array, where N is the number of emission lines defined in EMISSION_ SETUP_FILE, whithin the wavelength range loglam_gal. It specifies the intensity (corrected for reddening) of the emission lines. The intensities of lines in a multiplet are constrained by the flux ratio defined in the EMISSION_ SETUP_FILE (see Table 2.5).
gas_fluxes	N elements array, where N is the number of emission lines defined in EMISSION_ SETUP_FILE, whithin the wavelength range loglam_gal. It specifies the fluxes (corrected for reddening) of the emission lines. The intensities of lines in a multiplet are constrained by the flux ratio defined in the EMISSION_ SETUP_FILE (see Table 2.5).

gas_ew	N elements array, where N is the number of emission lines defined in EMISSION_ SETUP_FILE, whithin the wavelength range loglam_gal. It specifies the equivalent widhts (corrected for reddening) of the emission lines. The intensities of lines in a multiplet are constrained by the flux ratio defined in the EMISSION_ SETUP_FILE (see Table 2.5). Equivalent widths are computed by comparing the emission line flux with the median flux of the stellar continuum in the spectral region $\lambda_i - 10 \cdot FWHM_i < \lambda < \lambda_i - 5 \cdot FWHM_i$ and $\lambda_i + 5 \cdot FWHM_i < \lambda < \lambda_i + 10 \cdot FWHM_i$, where λ_i is the central wavelength of the i-th emission line, and FWHM_i is its measured FWHM (intrinsic plus instrumental).
gas_intens_err	N elements array, error on gas_intens.
gas_fluxes_err	N elements array, error on gas_fluxes.
gas_ew_err	N elements array, error on gas_ew.
OPTIONAL OUTPUTS	
bestfit	N elements array containing the best fit model (stars + gas), defined over the loglam_gal vector.
. ERROR	8 elements array, containing the errors on the kinematic parameters. ERROR [0]: error on the stellar velocity (km/sec). ERROR [1]: error on the stellar velocity dispersion (km/sec). ERROR [2]: error on the stellar h3 Gauss-Hermite moment. ERROR [3]: error on the stellar h4 Gauss-Hermite moment. ERROR [4]: error on the stellar h5 Gauss-Hermite moment (not used). ERROR [5]: error on the stellar h6 Gauss-Hermite moment (not used). ERROR [6]: error on the gas mean velocity (km/sec). ERROR [7]: error on the gas mean velocity dispersion (km/sec).
reddening	1 or two elements array that contain the best fit reddening values for star (1 element array) and for the gas (2 elements array). Input values will be overwritten.
err_reddening	errors associated to reddening, if fitted.
fitted_pixels	array. Indices of the good pixels used in the gandalf fit.
status	bolean. If 0, the ppxf fit did not succeeded. The gandalf fit is still carried on, and the stellar kinematics are fixed to the starting guesses. If 1, the ppxf fit converged.

3.5 mdap_ppxf.pro

This module fits the input galaxy spectrum with a series of stellar templates to get the best fit values of stellar kinematics, and stellar weights.

The original code is by M. Cappellari and E. Emsellem (Cappellari & Emsellem, 2004). The following modifications to the original code have been performed to adapt it to the MaNGA DAP workflow.

- Inclusion of the optional inputs range_v_star, and range_s_star to set the boundaries (with respect the stating guesses) where the best fit stellar velocity and velocity dispersion should be searched.
- Inclusion of the keyword /fix_star_kin, to fix the stellar kinematics to the input values.
- Inclusion of the optional keyword /external_library, to run the fortran version of the BVLS module.

Table 3.4 lists the inputs and outputs of mdap_ppxf.pro (from the original ppxf.pro description, with the addition of the new implemented keywords).

Table 3.4: Inputs and outputs parameters of mdap_ppxf.pro

INPUTS	
TEMPLATES	N elements 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 ¿= the number of galaxy pixels. - 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 PPXF, 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.

GALAXY	N elements vector containing 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. - For high redshift galaxies, one should bring the spectra close to the restframe wavelength, before doing the PPXF 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 (Warning: not yet implemented in the DAP). - GALAXY can also be an array of dimensions `GALAXY[nGalPixels,2]` containing two spectra to be fitted, at the same time, with a reflection-symmetric LOSVD. This is useful for spectra taken at point-symmetric spatial positions with respect to the center of an equilibrium stellar system. For a discussion of the usefulness of this two-sided fitting see e.g. Section 3.6 of Rix & White (1992, MNRAS, 254, 389). (Warning: this feature has not been tested in the context of the DAP workflow) - IMPORTANT: 1) For the two-sided fitting the VSYST keyword has to be used. 2) Make sure the spectra are rescaled to be not too many order of magnitude different from unity, to avoid over or underflow problems in the calculation. E.g. units of erg/(s cm^2 Å) may cause problems!
NOISE	N elements vector containing the 1*sigma error (per pixel) in the galaxy spectrum. If GALAXY is a Nx2 array, NOISE has to be an array with the same dimensions. - IMPORTANT: the penalty term of the pPXF 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	two elements vector `[velStart, sigmaStart]` with the initial estimate for the velocity and the velocity dispersion in km/s. - Unless a good initial guess is available, it is recommended to set the starting `sigma >= 3*velScale` in km/s (i.e. 3 pixels). In fact when the LOSVD is severely undersampled, and far from the true solution, the chi^2 of the fit becomes weakly sensitive to small variations in sigma (see pPXF paper). In some instances the near-constancy of chi^2 may cause premature convergence of the optimization. - In the case of two-sided fitting a good starting value for the velocity is velStart=0.0 (in this case VSYST will generally be nonzero). Alternatively on should keep in mind that velStart refers to the first input galaxy spectrum, while the second will have velocity -velStart.
KEYWORDS	
BESTFIT	a named variable to receive a vector with the best fitting template: this is a linear combination of the templates, convolved with the best fitting LOSVD, with added polynomial continuum terms.
BIAS	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: the solution (including [V,sigma]) 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 λ 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.
/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.
DEGREE	degree of the *additive* Legendre polynomial used to correct the template continuum shape during the fit (default: 4). Set DEGREE = -1 not to include any additive polynomial.

ERROR	a named variable that will contain a vector of *formal* errors (1*sigma) for the fitted parameters in the output vector SOL. This option can be used when speed is essential, to obtain an order of magnitude estimate of the uncertainties, but we *strongly* recommend to run Monte Carlo simulations to obtain more reliable errors. In fact these errors can be severely underestimated in the region where the penalty effect is most important (sigma < 2*velScale). - These errors are meaningless unless Chi^2/DOF$\sim$1 (see parameter SOL below). However if one *assume* that the fit is good, a corrected estimate of the errors is: errorCorr = error*sqrt(chi^2/DOF) = error*sqrt(sol[6]). - IMPORTANT: when running Monte Carlo simulations to determine the error, the penalty (BIAS) should be set to zero, or better to a very small value. See Section 3.4 of Cappellari & Emsellem (2004) for an explanation.
GOODPIXELS	integer vector containing 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. - IMPORTANT: in all likely situations this keyword *has* to be specified.
LAMBDA	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.
MDEGREE	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	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. Warning: The DAP workflow fits only moments up to h4.

/OVERSAMPLE	Set this keyword to oversample the template by a factor 30x before convolving it with a well sampled LOSVD. This can be useful to extract proper velocities, even when sigma ¡ 0.7*velScale and the dispersion information becomes totally unreliable due to undersampling. IMPORTANT: One should sample the spectrum more finely is possible, before resorting to the use of this keyword!
Warning: not tested in the DAP workflow. /PLOT	set this keyword to plot the best fitting solution and the residuals at the end of the fit. Warning: DO NOT USE IN THE DAP WORKFLOW.

POLYWEIGHTS	vector with 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.

/QUIET	set this keyword to suppress verbose output of the best fitting parameters at the end of the fit.
REDDENING	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. - IMPORTANT: The MDEGREE keyword cannot be used when REDDENING is set. Warning: This keyword is NEVER used in the DAP workflow: the reddening is measure usiong Gandalf.

REGUL	(Warning: not tested in the DAP workflow) If this keyword is nonzero, the program applies second-degree linear regularization to the WEIGHTS during the PPXF 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: (i) 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)); (ii) 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: 1. Perform an un-regularized fit (REGUL=0) and then rescale the input NOISE spectrum so that Chi^2/DOF = Chi^2/N_ELEMENTS(goodPixels) = 1. This is achieved by rescaling the input NOISE spectrum as `NOISE = NOISE*sqrt(Chi`^{2}`/DOF) = NOISE*sqrt(SOL[6])`; 2. Increase REGUL and iteratively redo the pPXF fit until the Chi^2 increases from the unregularized Chi^2 = N_ELEMENTS(goodPixels) value by $\Delta\mathrm{Chi}^2$ = 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).

SKY	Warning: not tested in the DAP. vector containing the spectrum of the sky to be included in the fit, or array of dimensions SKY[nPixels,nSky] containing different sky spectra to add to the model of the observed GALAXY spectrum. The SKY has to be log-rebinned as the GALAXY spectrum and needs to have the same number of pixels. - The sky is generally subtracted from the data before the PPXF fit. However, for oservations very heavily dominated by the sky spectrum, where a very accurate sky subtraction is critical, it may be useful *not* to subtract the sky from the spectrum, but to include it in the fit using this keyword.
VSYST	difference between the log-lam values of the wavelength of the galaxy and the template stars, expressed in km/sec.

WEIGHTS	a named variable to receive the value of the weights by which each template was multiplied to best fit the galaxy spectrum. The optimal template can be computed with an array-vector multiplication:
	
	T `EMP = TEMPLATES # WEIGHTS` (in IDL syntax)
	- When the SKY keyword is used WEIGHTS[0:nTemplates-1] contains the weights for the templates, while WEIGHTS[nTemplates:*] gives the ones for the sky. In that case the best fitting galaxy template and sky are given by:
	`TEMP = TEMPLATES # WEIGHTS[0:nTemplates-1]`
	`BESTSKY = SKY # WEIGHTS[nTemplates:*]`
	- When doing a two-sided fitting (see help for GALAXY parameter) *together* with the SKY keyword, the sky weights are allowed to be different for the left and right spectrum. In that case the output sky weights alternate between the first (left) spectrum and the second (right) spectrum.

/fix_star_kin	If set, the stellar kinematics are not fitted. The return value is that of the starting guesses.
range_v_star	2 elements array]. It specifies the boundaries for the stellar best fit velocity (in km/sec). Default: starting_guess $\pm$ 2000 km/sec.
range_s_star	2 elements array]. It specifies the boundaries for the stellar best fit velocity dispersion (in km/sec). Default: $21 < \sigma < 499$ km/sec.
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.
OUTPUTS	

SOL	seven elements vector containing in output the values of `[Vel, Sigma, h3, h4, h5, h6, Chi`^{2}`/DOF]` 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. Warning: in the DAP workflow, only moments up to h4 are fitted.
	- I hardcoded the following safety limits on the fitting parameters:
	• Vel is constrained to be ±2000 km/s from the first input guess
	• $velScale/10 < Sigma <$ 500 km/s
	• $-0.4 < [h3, h4, ...] < 0.4$ (limits are extreme value for real galaxies)
	- IMPORTANT: if Chi^2/DOF is not 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 7+MDEGREE elements `[Vel, Sigma, h3, h4, h5, h6, Chi`^{2}`/DOF, cx1, cx2, ..., cxn]`, where cx1, cx2, ..., cxn 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]`

3.6 `mdap_gandalf.pro`

This main module is an implementation of the original Gandalf program by Sarzi et al. (2006). Modifications to the original procedure include:

- It takes into account different values of the instrumental velocity dispersion for each emission lines (i.e. INST_DISP is function of wavelenght).
- It is possible to fix the gas kinematics to the starting guesses (fix_gas_kin keyword), or specify the best fit boundaries (range_v_gas and range_s_gas keywords).
- Inclusion of the optional keyword /external_library, to run the fortran version of the BVLS module.
- convolution is performed with the same utility module used by pPXF: mdap_ppxf_convol_fft.pro

Table **??** lists the inputs and outputs of the `mdap_gandalf.pro` module.

Table 3.5: Inputs and outputs parameters of mdap_gandalf.pro

INPUTS	
TEMPLATES	vector containing the spectrum of a single template star or array of dimensions nPixels_Templates x nTemplates containing different templates to be optimized during the fit of the kinematics. nPixels_Templates has to be >= the number of pixels sampling the galaxy spectrum nPixels_Galaxy.
GALAXY	nPixels_Galaxy elements vector containing the spectrum of the galaxy to be measured. Both star and the galaxy spectra HAVE to be logarithmically rebinned on a natural ln-base (or in log10, see LOG10 keyword). Warning: the MaNGA DAP uses a natural ln-base.
NOISE	nPixels_Galaxy elements vector containing the 1*sigma error of the emission spectrum. If this is not available an array of constant unity values should be passed.
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.
SOL	on INPUT it must be a vector containing the stellar ; kinematics needed to convolve the input stellar spectrum or ; template library (V, σ, h3, h4, h5, and h6). On OUTPUT it will contain the results of the gas fit with weights assigned to the multiplicative polynomial appended.
EMISSION_SETUP	A structure containing the index, the name, and the wavelength of the fitted emission lines. It must also contain the starting/input values for the line amplitudes, velocities, and widths of the lines, as well as keywords specifying whether each line is a part of a doublet, and whether its position and width fit freely, to be hold at its input values, or to be tied that of another line.

L0_GAL	Float. the ln-lambda value corresponding to the starting pixel in the galaxy spectrum.
L0_STEP.	Float. the ln-lambda step of corresponding to the pixels sampling the galaxy spectrum. Warning: Wavelengths are assumed to be in Angstrom.
KEYWORDS	
DEGREE	Integer. degree of the Legendre polynomial used for correcting the template continuum shape during the fit (default: -1). Warning: the DAP uses DEGREE = -1.
MDEGREE	Integer. degree of the Legendre polynomial used for correcting the template continuum shape during the fit (default: 0). This correction is MULTIPLICATIVE. Warning: the DAP sets it automatically to 0 if the REDDENING is fitted.
GOODPIXELS	integer vector containing the indices of the pixels in the galaxy spectrum (in increasing order) that will be included in the fit. IMPORTANT: in all likely situations this keyword *has* to be specified.
INT_DISP	Nlines elements vector, where N_lines is the number of emission li nes defined in the user-input emission line set-up file (see **??**). It contains the instrumental velocity dispersion for each line (in km/sec), as function of its observed wavelength (gas velocity starting guess is used to calculate the observed wavelength).
LOG10: allows to deal with data that have been logarithmically rebinned in lambda, using a base 10. Warning: the DAP uses natural log base.	
REDDENING	1 or 2 elements array. It allows to include in the fit the effect of reddening by dust, by specifying a single $E(B-V)$ guess for extinction, or a two-element array of $E(B-V)$ guesses. A single guess will trigger the use of a single-screen dust model, affecting the entire spectrum, whereas passing a two-elements array of guesses will add a second dust component affecting only the nebular fluxes. This second option is recommended only when recombination lines are clearly detected, so that a temperature-dependent prior (e.g. the decrement of the Balmer lines) on their relative intensity can be use to constrain such a second, internal component.
L0_TEMPL	the ln-lambda value corresponding to the starting pixel in the template spectra. Needed if using the REDDENING keyword and the DUST_CALZETTI function.

FOR_ERRORS	A keyword specifying whether we wish errors to be properly estimated. Warning: in the DAP this keyword is always activated.
/QUIET:	set this keyword to mute the output.
/PLOT:	set this keyword to plot the best fitting solution at the end of the fit. Warning: do not USE it in the DAP.
OUTPUT PARAMETERS	
SOL	A 4xNlines vector containing the values of the best fitting flux, amplitude, Velocity, and velocity dispersion for each emission line. If the keyword INT_DISP has been set the velocity dispersion is already the intrinsic one, otherwise it will be the observed width of the line.
BESTFIT	a named variable to receive a vector containing sum of best fitting stellar (convolved by the input LOSVD) and emission-line Gaussian templates. This model include bending of the stellar templates by the best fitting multiplicative polynomials, and of additive polynomials, if specified.
EMISSION_ TEMPLATES:	a named variable to receive an [nPixels_Galaxy, Nlines] array containing the best fitting emission-lines templates. The emission spectrum can be obtained from this array by simply doing total(emission_templates,2).
WEIGHTS	a named variable to receive the value of the weights by which each template was multiplied to best fit the galaxy spectrum.
ERROR	a named variable that will contain a vector of formal errors (1 sigma) for the parameters in the output vector SOL. If the FOR_ERRORS keyword IS NOT specified no errors for the line amplitudes and fluxes will be returned and the uncertainties on the position and width of the lines should be regarded only as order of magnitude estimates. If the FOR_ERRORS keyword IS specified (Default in the DAP), a second fully non-linear fit of all the emission-line parameter will provide correct estimates for the errors on all emission-line parameters. Still, Keep in mind that these errors are meaningless unless $Chi^2/DOF \sim 1$. If a constant noise spectrum is provided, the formal errors will be automatically rescaled under the assumption the model is a good representation of the data.
/fix_gas_kin	If set, the gas kinematics are not fitted. The return value is that of the starting guesses.
range_v_gas	2 elements array]. It specifies the boundaries for the gas best fit velocity (in km/sec). Default: starting_guess $\pm$ 2000 km/sec.
range_s_gas	2 elements array]. It specifies the boundaries for the gas best fit velocity dispersion (in km/sec). Default: $21 < \sigma < 499$ km/sec.

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.

3.7 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(λ), 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)

3.8 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 3.7.

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

Warning: "D4000" and "TiO0p89" (i.e. TiO0.89) indices are defined as the ratio of the flux in the red and blue pseudocontinua. This alternative configuration is hardwired in mdap_measure_indices.pro (Section 2.5.1).

Table 3.7: 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.

3.9 mdap_kinemetry.pro

This main module implements the kinemetry.pro module by Kajinovic et al. 2008) to measure the rotation curve (stars and gas), position angle, and axial ratio and amplitude of inflows/outflows.

Table 3.8 lists the inputs and outputs of mdap_kinemetry.pro (from the original module by D. Krajinovic).

Table 3.8: Inputs and outputs parameters of mdap_kinemetry.pro

INPUTS	
XBIN	1D array with X coordinates describing the map.
YBIN	1D array with Y coordinates describing the map.
MOMENT	1D array with kin.moment (e.g. velocity) values at XBIN, YBIN positions.
OPTIONAL INPUTS KEYWORDS	
NTRM	scalar specifying the number of terms for the harmonic analysis of the profile extracted along the best fitting ellipse. Default value is 6 odd terms, which means the following terms will be used in the expansion: a1, b1, a3, b3, a5 and a5, or the terms corresponding to sin(x), cos(x), sin(3x), cos(3x), sin(5x), cos(5x). Warning: the DAP uses NTRM=2.
ERROR	1D array with errors to VELBIN values. If this keyword is specified then the program will calculate formal (1sigma) errors of the coefficients which are returned in ER_PA, ER_Q (determined by MPFIT), ER_CF (determined by SVD) variables. If IMG keyword is set ERROR has to be a 2D array. If if is not supplied, it is creatred as 2D arrays with all values equal to 1.
SCALE	scalar specifying the pixel scale on the map. If not set, the SAURON pixel scale (0.8 arcsec) is assumed. Warning: DAP uses scale=1.

IMG	2D array containing an image. This keyword was designed specifically for surface photometry (analysis of the zeroth moment of the LOSVD), to increase the spead of calculations, but if kinematic map is large (and has many pixels) it can be also passed through this keyword and analysis will be quicker. To use kinemetry for photometry it is also necessary to set keyword EVEN and it can be useful to use NTRM=10 in order to measure disky/boxy deviations (4th terms). Images are very different from current kinematic maps. They are much larger and usually not binned, so make regular 2D grids. This is the reason to treat the kinematic maps and images in a different way. If IMG is set, the treatment follows the ideas of Jedrzejewsky (1987): inner parts of (small radii, rad ¡ 40 pixels) are interpolated while outer parts are binned in elliptical sectors (64 in angle to increase the signal and of thicknes equal to 10% of the given radius). When IMG is used center can be also fitted (currently only in photometry, so if keyword EVEN is not set - which usually means one is analysing a velocity map - center is not fitted). An estimate of the center is given through XC and YC keywords. ERROR should be a 2D array of the same size as IMG. When IMG is used, VELCIRC and VELKIN keywords contain reconstructed images. It is assumed that image coordiantes are in pixels (not physical units). Keyword SCALE is automatically set to 1. If this is not the case, set SCALE to a required number. In order to be compatible with previous versions of kinemetry XBIN, YBIN and MOMENT still have to be passed but they can be dummmy 1D variables, unless certain image areas are masked (as bad pixels). IF keyword BADPIX is used, XBIN and YBIN should be 1D arrays with the actual coordinates of the IMG. One can use the following set of lines to make the arrays:
	`s=size(img)`
	`n=s[1]*s[2]`
	`yy=REPLICATE(1,s[1])#(indgen(s[2]))`
	`xx=(indgen(s[1]))#REPLICATE(1,s[2])`
	`xbin=REFORM(xx, n)`
	`ybin=REFORM(yy, n)`

X0	an estimate of the X coordinate of the center (in pixels). If not given X0=0. For accurate determination of the center and other ellipse parameters at small radii it is important the ellipse includes the center of the galaxy.
Y0	an estimate of the Y coordinate of the center (in pixels). If not given Y0=0.
FIXCEN	keyword, if set center will be fixed and no attempt will be made during the harmonic decomposition to find new center. Center is fixed to X0 and Y0. This keyword is optional only for photometry (or even moments in general). For ODD moments, center is fixed always.

NRAD	scalar specifying the number of radii along which kinemetry should be run. IF not specified, NRAD=100. Kinemetry will stop when the edge of the map is encountered and NRAD is not necessary achived. To force kinemetry to do all radii, relax condition in keyword COVER.
NAME	name of the object (used by VERBOSE keyword and for internal plotting).
PAQ	2 element or 2*NRAD element vector specifying position angle (PA) and flattening (q) of the ellipses in that order (kept constant). It is possible to specify a set of PA and q values (that correspond to given radii (see RADIUS keyword)), for which one wants to get the Fourier coefficients. In this case PAQ should be set as follows: PAQ=[PA1, Q1, PA2, Q2...., PAnrad, Qnrad] It can be also used as an initial condition for determination of ellipses. In this case, it should be called together with /NOGRID keyword (currently implemented only for photometry). IF PAQ keyword is used to define ellipses along which harmonic decomposition is made, then keyword NOGRID should not be used. In this case center is fixed (and should be defined via X0 and Y0 keywords if IMG keyword is used).
NOGRID	keyword, if set it bypasses the direct minimisation via a grid in PA and Q values. It should be used together with PAQ parameters, when a good estimate of PA and Q are passed to the program, but not if PAQ serve to pre-define ellipses for harmonic decomposition. It is desigend with photometry in mind, where the problem usually has only one well defined minimum (in PA,Q plane). It speeds up the calculation, but for the higher kinematic moments it is not as robust and it is advised not to be used (first use the grid to find the best fit PA and Q values.).
NPA	scalar specifying the number of PA used to crudely estimate the parameters of the best fit ellipse before entering MPFIT. Default value is 21. To speed up the process and for quick tests it is useful to use a small number (e.g 5). Complicated maps may require a bigger number (e.g. 41).
NQ	scalar specifying the number of q used to crudely estimate the parameters of the best fit ellipse before entering MPFIT. Default value is 21. To speed up the process and for quick tests it is useful to use a small number (e.g 5). Complicated maps may require a bigger number (e.g. 41).
RANGEQ	2 element vector specifying the min and max value for flattening Q. Default values are 0.2 and 1.0.
RANGEPA	2 element vector specifying the min and max value for position angle PA. Default values are -90 and 90 (degrees).

BADPIX	1D array containing indices of pixels which should not be used during harmonic fits. This keyword is used when data are passed via IMG. It is usefull for masking stars and bad pixels. When used, XBIN and YBIN should be real coordinates of the pixels of IMG (see IMG for more details). The bad pixels are passed to the routine which defines/selects the ellipse coordiantes (and values) to be fitted, and they are removed from the subsequent fits. All pixels of the ellipse which are 2*da from the bad pixels are removed from the array, where da is the width of the ring.
/ALL	If this keyword is set then the harmonic analysis of the rings will include both even and odd terms. If this keyword is set, and NTRM = n then the following terms are used in the expansion: a1, b2, a2, b2, a3, b3,...., an, bn (or coeffs nex to: sin(x), cos(x), sin(2x), cos(2x), sin(3x), cos(3x),...,sin(nx),cos(nx)).
/EVEN	set this keyword to do kinemetry on even kinematic moments. In this case, kinemetry reduces to photometry and the best fitting ellipse is obtained by minimising a1, b1, a2, b2 terms. When this keyword is set, keyword /ALL is automatically set and NTRM should be increased (e.g. NTRM=10 will use the following terms in the expansion: a1, b2, a2, b2, a3, b3, a4, b4 (or coeffs. next to sin(x),cos(x),sin(2x), cos(2x),sin(3x), cos(3x),sin(4x),cos(4x))).
/VSYS	if this keyword is set the zeroth term (a0) is not extracted. (for odd moments).This might be useful for determinatio of rotation curves.One can first run kinemetry without setting this keyword to find the systemic velocity (given as cf[*,0]). Then subtract the systemic velocity form the velocity map and re-run kinemetry with /vsys set. In this case the zeroth terms will be zero. For completeness, it is also possible to input VSYS, e.g. VSYS=10. The zeroth term will not be calculated, but it will be set to 10 in output. Given that Fourier terms are orthogonal, it should not be necessary to set this keyword in general.
RING	scalar specifying desired radius of the first ring. Set this keyword to a value at which the extraction should begin. This is useful in case of ring-like structures sometimes observed in HI data.
RADIUS	1D array with values specifying the lenght of the semi-major axis at which the data (kin.profile) should be extracted from the map for the kinemetric analisys. The values should be in pixel space (not in physical units such as arcsec). If this keyword is set, the values are coopied into the output variable: RAD.
COVER	Keyword controling the radius at which extraction of values from the map stops. Default value is 0.2, meaning that if less than 20% of the points along an ellipse are are not present, the program stops.

/BMODEL	If this keyword is set, a model moment map is constructed. This keyword should be set together with VELCIRC and VELKIN keywords, which will contain the reconstructed map, using the first dominant term and all terms, respectively. IF IMG keyword is used, the outputs are 2D images, otherwise BMODEL reconstructs the map at each input position XBIN,YBIN. If BMODEL is not set VELCIRC and VELKIN will contain reconstructed values at the positions of XELLIP and YELLIP.
/PLOT	If this keyword is set, diagnostic plots are shown for each radii: - the best ellipse (overploted on kin.moment map). If IMG keyword is set, the image is scaled to the size of the ellipse. The centering of the overplotted ellipse is good to 0.5 pixels so for small radii (r ¡ a few pixels) it is possible that the position of the center of the overplotted ellipse is not on the brightes pixel. - PA-Q grid with the position of the minimum (green diamond) for the parameters of the best fit ellipse determined by MPFIT, where the initial (input to MPFIT) values of PA and Q are presented by the grid of dots and the colours show the Chi2 square contours (linearly interpolated between the PA,Q points), - fit to kin.profile (white are the DATA, red is the FIT, where FIT is given by a0+b1*cos(x) for odd, and a0 for even moments), - residuals (DATA - FIT), and overplotted higher order terms (green: a1,a3 and b3, red: a1,a3,b3,a5 and b5; for the /EVEN case - green: a1, b1, a2, b2, red: a1, b1, a2, b2, a4, b4) Warning: do not use in the DAP.
/VERBOSE	set this keyword to print status of the fit on screen including information on: • Radius - number of the ring that was analysed • RAD - radius of the analysed ring (if SCALE is passed it is given in the same units, otherwise in pixels) • PA - position angle of the best fitting ellipse • Q - flattening of the best fitting ellipse • Xcen - X coordinate of the centre of the ellipse • Ycen - Y coordinate of the centre of the ellipse • Number of ellipse elements - number of points to which the data points in the ring are sampled before derivation of the best fit parameters and harmonic analysis. It varies between 20 and 64 (or 100 in non IMG model) depending on the ring size, giving a typical sampling of 5.6 (3.6) degrees.
OUTPUT	
RAD	1D array with radii at which kin.profiles were extracted.

PA	1D array with position angle of the best fitting ellipses, PA is first determined on an interval PA=[-90,90], where PA=0 along positive x-axis. Above x-axis PA $>$ 0 and below x-axis Pa $<$ 0. PA does not differentiate between receding and approaching sides of (velocity) maps. This is transformed to the usual East of North system, where the East is the negative x-axis, and the North is the positive y-axis. For odd kin.moments PA is measured from the North to the receding (positive) side of the galaxy (which is detected by checking the sign of the cos(theta) term. For the even terms it is left degenerate to 180 degrees rotation.
Q	1D array with flattening of the best fitting ellipses (q=1-ellipticity), defined on interval q=[0.2,1]
CF	2D array containing coefficients of the Fourier expansion for each radii cf=[Nradii, Ncoeff]. For example: a0=cf[*,0], a1=cf[*,1], b1=cf[*,2]....
bf OPTIONAL OUTPUT KEYWORDS	
VELKIN	1D array of reconstructed kin.moment using NTRM harmonic terms at positions XBIN,YBIN, obtained by linear interpolation from points given in XELLIP and YELLIP keywords (if BMODEL keyword is used, otherwise at positions XELLIP and YELLIP.
VELCIRC	1D array containinng 'circular velocity' or a0 + b1*cos(theta) at positions XBIN, YBIN (velcirc = a0, in case of EVEN moments), obtained by linear interpolation from points given in XELLIP and YELLIP keywords (if BMODEL keyword is used, otherwise at positions XELLIP and YELLIP.
GASCIRC	1D array containing circular velocity or Vcirc=cf[*,2]*cos(theta) at positions XBIN and YBIN, obtained for fixed PA and q. PA and q are taken to be median values of the radial variation of PA and q. IF keyword PAQ is used than GASCIRC give the circular velocity (no systemic velocity) for the median values of PA, Q values. Note that this is different from VELCIRC (also VELKIN) which is obtained on the best fitting ellipses and also includes Vsys (cf[*,0]) term. This keyowrd is useful for gas velocity maps, if one wants to obtain a quick disk model based on the circular velocity.
ER_PA	1D array of 1sigma errors to the ellipse position angle.
ER_Q	1D array of 1sigma errors to the ellipse axial ratio.
ER_CF	2D array containing 1 sigma errors to the coefficients of the Fourier expansion for each radii.
XELLIP	1D array with X coordintes of the best fitting ellipses.
YELLIP	1D array with Y coordintes of the best fitting ellipses.
XC	the X coordinate of the center (in pixels). If X0 not fit XC=0.
YC	the Y coordinate of the center (in pixels). If Y0 not fit YC=0.

Chapter 4

Utility modules

4.1 mdap_convol_sigma.pro

This function convolves an input vector (e.g. stellar spectrum), defined over a vector x (e.g. wavelength) with a Gaussian kernel with standard deviation that depends on x.

Tables 4.1 lists the Inputs and outputs parameters of `mdap_convol_sigma.pro`.

Table 4.1: Inputs and outputs parameters of mdap_convol_sigma.pro

INPUTS	
x	[1D array] Values for which "vector" is defined Units: pixels (or angstrom, but with constant ang/step sampling).
vector	[1D array] vector to be convolved. Same elements of "x".
x_sigma	[1D array] Values for which "sigma" is defined/
sigma	[1D array] Values of the dispersion of the convolving Gaussian function. Same elements of "x_sigma" Units: same as X.
RETURN VALUE	
RETURN VALUE	Convolved vector, same elements of "vector"

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 4.2: 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