The MANGA Data Analysis Pipeline: a prototype

L. Coccato & MANGA team

January 24, 2014

Contents

Chapter 1

Introduction

1.1 Data Analysis Pipeline: overview

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

High level data products are:

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

Model dependent data products are:

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

1.1.1 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 1.1 list a summary of the blocks, interfaces and main modules distribution.

The input datacube is read in block 1 and information (vectors with galaxy spectra, errors, wavelenght, and spatial information) are passed to the block 2 for spatial binning. Three spatial binnings are foreseen, depending on the scientific requirements. The current DAP version uses the Voronoi binning scheme, (as implemented in IDL by Cappellari & Copin 2003), as main module for the spatial binning task.

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

The log-sampled spectra are then passed to block 4 for spectroscopic measurements. Three fits are performed in block 4, one for each spatial sampling defined in block 2. Before each fit, the log-sampled Galaxy spectra are corrected for Milky Way extinction(input parameter). Results from the first execution will be used to constrain the fit of the second fit, and soforth. The current version uses i) sgandalf.pro (a modified version of ppxf.pro by Cappellari & Emsellem 2004, and spectral decomposition by Coccato et al. 2011); and ii) Calzetti et al. (2000) formulas for reddening correction as main modules in block 4 (See Section 3.3.2 for further details). 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.

1.1.2 DAP inputs and outputs

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

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

BLOCK	Interface modules	Main and utility modules
Block 1	mdap_read_datacube	mdap_calculate_spectrum_sn
Block 2	mdap_spatial_binning	mdap_voronoi_2d_binning mdap_calibrate_sn
Block 3	mdap_log_rebin[1]	mdap_do_log_rebin mdap_convol_sigma
Block 4	mdap_spectral_fitting[3] mdap_create_starting_guesses	mdap_calculate_spectrum_sn mdap_bvls mdap_dust_calzetti mdap_get_losvd `mpfit package` mdap_sgandalf mdap_gandalf_wrapf[2] mdap_gandalf[2] mdap_ppxf[2]
Block 5	mdap_measure_indices	mdap_read_indices_definitions[4] mdap_do_measure_index mdap_convol_sigma
Block 6	mdap_spatial_radial_binning mdap_spectral_fitting[3] mdap_measure_indices	mdap_calculate_spectrum_sn mdap_read_indices_definitions[4] mdap_do_measure_index mdap_convol_sigma mdap_dust_calzetti mdap_get_losvd `mpfit package` mdap_sgandalf mdap_gandalf_wrapf[2] mdap_gandalf[2] mdap_ppxf[2]
Block 7		
Block 8		
Block 9		

Notes: [1] Need auxiliary files: stellar templates. [2] Need auxiliary file: Emission lines definitions. [3] It runs either mdap_sgandalf, or the 3 routines package: mdap_gandalf_wrap plus mdap_gandalf plus mdap_ppxf. [4] Need auxiliary file: Absorption line index definitions.

To run the pipeline, the following files are needed.

- total_filelist.dat. 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 datacubes listed in the file total_filelist.dat.
- a set of stellar templates, in fits file format (see Section 2.3.1).
- emission_lines_setup_with_Balmer_decrement. A file containing the definitions of the emission lines to include in the fit (see Section **??**).
- absorption_line_indices_definition.dat. A file containing the definitions of the absorption line indices to measure (see Section 3.3.6).

The DAP is executed by the following IDL command line

$$\texttt{IDL} > \texttt{manga\_dap}, \texttt{i}, [/\texttt{check\_version}], [/\texttt{dont\_remove\_null\_templates}]$$

where `i` is the index number of the i-th datacube in total_filelist.dat, column 1 to analyse, $i = 0, N-1$.

/check_version performs the reduced version control on already existing output data, and avoid to redude again the data if the previous reduction is up to date.

/dont_remove_null_templates By default, the DAP speeds up the analysis by excluding some stellar templates that have zero weight (See Section 3.3.2. If this keyword is set, the speed up process is not activated.

As output, the DAP returns a multilayer fits file with all the relevant quantities ((`<datacube_name>_high_level.fits`), an idl session with all the session variables stored (`<datacube_name>_mdap_session.idl`). These outputs are stored and a log file (¡datacube_name¿_mdap.log) in the directory `results/ <datacube_name>/`, 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).bla bla
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.

Ext	Name	Description
5	binning 2 data	Measurements performed on the second binning scheme (stellar kinematics).
6	binning map 3	Location and geometry of the spatial bins of the third binning scheme.
7	binning 3 data	Measurements performed on the third binning scheme (gas kinematics and emission line properties).
8	binning map radial	Location and geometry of the radial binning scheme.
9	binning radial data	Measurements performed on the radial binning scheme (absorption line indices and stellar velocity dispersion)

1.1.3 Version control and tags

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.

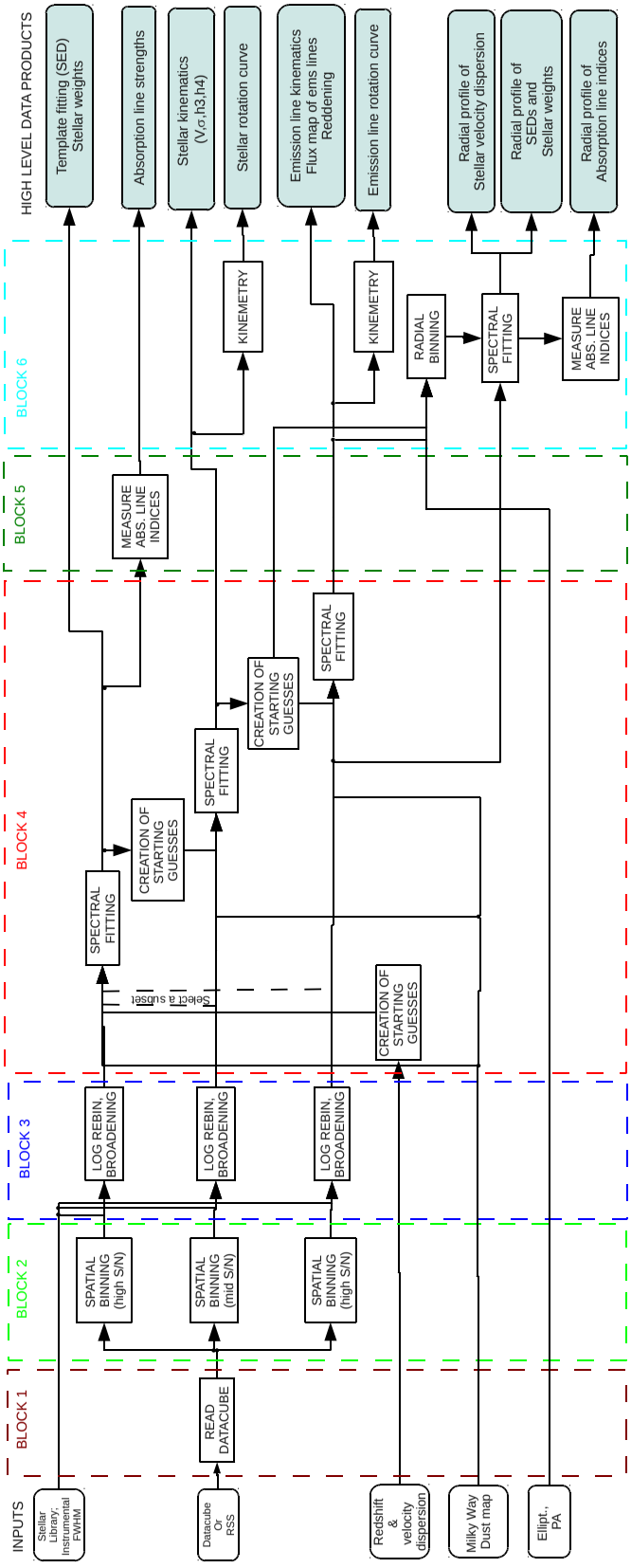

Figure 1.1: Workflow chart of the Data Analysis Pipeline (High Level Data products).

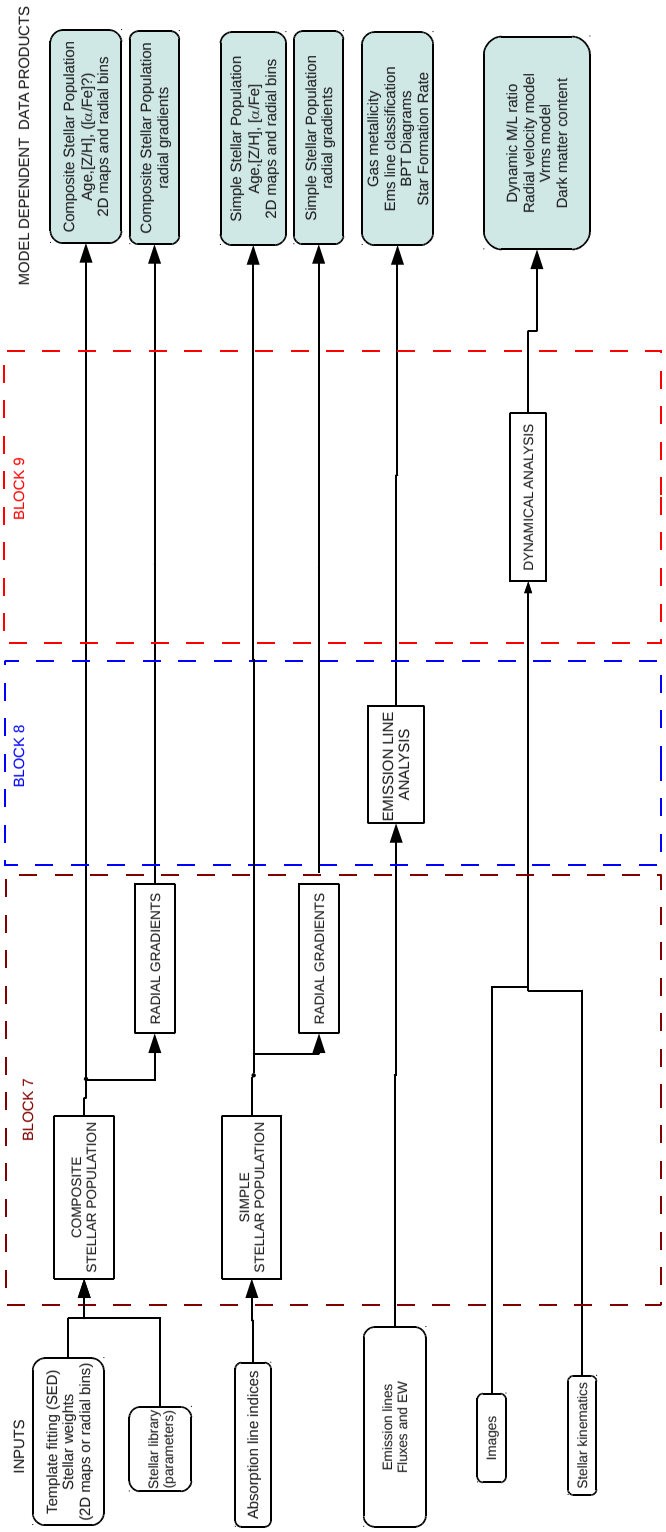

Figure 1.2: Workflow chart of the Data Analysis Pipeline (Model Dependent Data products).

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

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

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

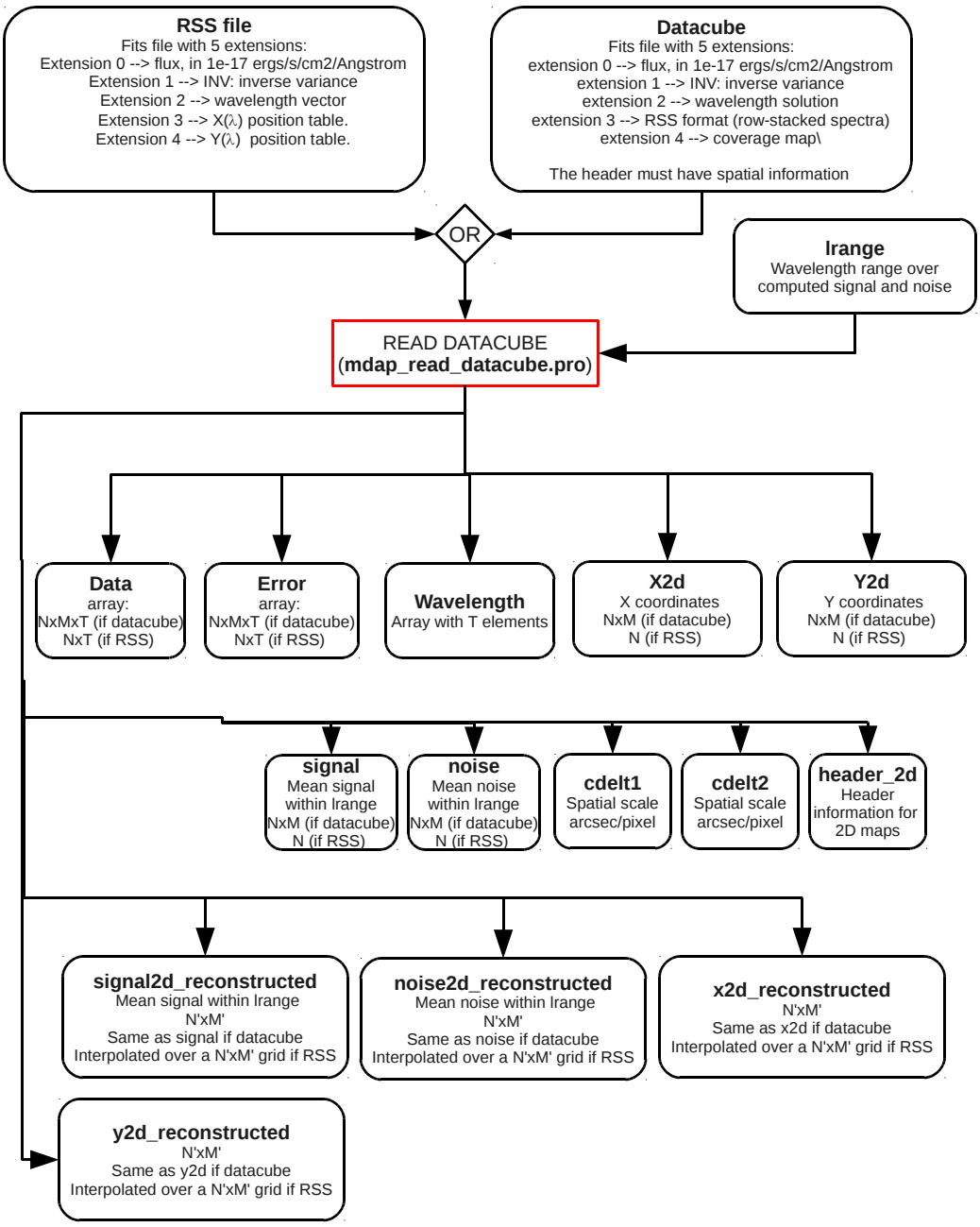

Figure 2.1: Workflow chart of the block 1 of the Data Analysis Pipeline. See Section 2.1.1 for a description of inputs and outputs.

2.1.1 `mdap_read_datacube.pro`

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

Table 2.1: Inputs and outputs parameters of mdap_read_datacube.pro

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

/keep_original_ step	If set, the wavelength output vector will be the same as the one define from the input fits file. The default is to re-construct (and therefore re-inrpolate the galaxy and error spectra) the output wavelength vector with constant ang/pixel step using the minimum ang/pixel step that is stored in the wavelength solution. For MANGA, it is suggested to set this keyword.
OUTPUTS	
data	Galaxy spectra. [NxMxT dbl array] in the case that the inputs are in datacube format, or [NxT dbl array] in the case the inputs are in RSS format. Spectra are resampled over the vector wavelength.
error	Errors associated to data. [NxMxT dbl array] in the case that the inputs are in datacube format, or [NxT dbl array] in the case the inputs are in RSS format. Spectra are resampled over the vector wavelength.
wavelength	[T dbl array]. Wavalenght value 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]).
x2d	Array containing the x coordinates in arcseconds (0 is the center of the field of view). [NxM dbl array] in the case that the inputs are in datacube format, or [N dbl array] in the case the inputs are in RSS format.
y2d	Array containing the y coordinates in arcseconds (0 is the center of the field of view). [NxM dbl array] in the case that the inputs are in datacube format, or [N dbl 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). [NxM dbl array] in the case that the inputs are in datacube format, or [N dbl 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.
noise	Mean galaxy error per Å, obtained considering all the wavelength range (or only the range specified by lrange). Calculation is done on original spectra, not those resampled over the vector wavelenght. [NxM dbl array] in the case that the inputs are in datacube format, or [N dbl 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.
cdelt1	[double]. Spatial sampling along x direction (arcsec/pixel). If inputs are in RSS format, it is set to 0.5.
cdelt2	[double]. Spatial sampling along y direction (arcsec/pixel). If inputs are in RSS format, it is set to 0.5.

header2d	[str array]. The header for output two-dimensional maps produced by the DAP.
OPTIONAL OUTPUTS	
version	string specifying the module version. If requested, the module is not execute and only version flag is returned.
x2d_ reconstructed	[NxM array] If input is in DATACUBE format, x2d_ reconstructed = x2d. [N'xM' 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 array] If input is in DATACUBE format, y2d_reconstructed = y2d. [N'xM' 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 array] If input is in DATACUBE format, signal2d_reconstructed = signal. [N'xM' 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 array] If input is in DATACUBE format, 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.

Future developments

The following items needs to be implemented:

- Implement with new format of input datacube.
- Allow for identification and removal of foreground stars.

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 pixel. 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 pixel. 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 pixel. 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 modules are: mdap_voronoi_2d_binning.pro (Section 3.2) and mdap_calibrate_sn.pro (Section 3.3).

mdap_voronoi_2d_binning.pro uses an imlementation of the Voronoi Binning technique developed by **?** (see Section 3.2). 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 main 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.

Figure 2.2 illustrates the flowchart of block 2.

future implementations

Possible implementations are:

- include the possibility to perform a radial and azimuthal binning, regardless of the signal-to-noise.

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

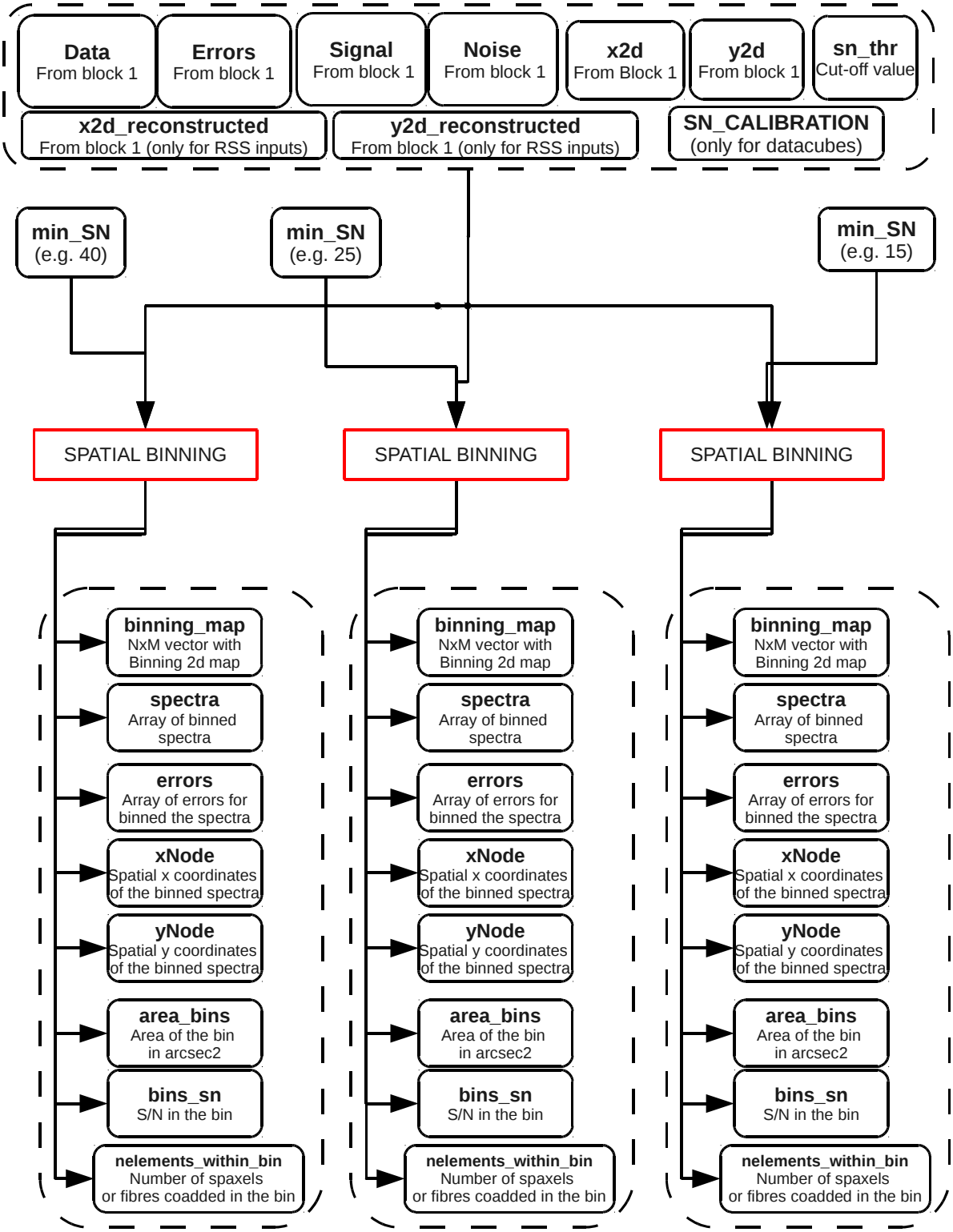

Figure 2.2: Workflow chart of the block 2 of the Data Analysis Pipeline.

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 to combine adjacent spectra in the field of view till a minimum S/N is reached (see Section 3.2).

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.

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: `t = SN_est^SN_CALIBRATION[0] / sqrt(n_elements(Nbin))` `SN_real = poly(t,SN_CALIBRATION[1:*])`, where `Nbin` is the number of spectra added in that spatial bin.
INPUT KEYWORDS	
/plot	If set, some plots on X11 terminal will be shown. Not suggested if the task is launched remotely.
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 3.3.1). 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).

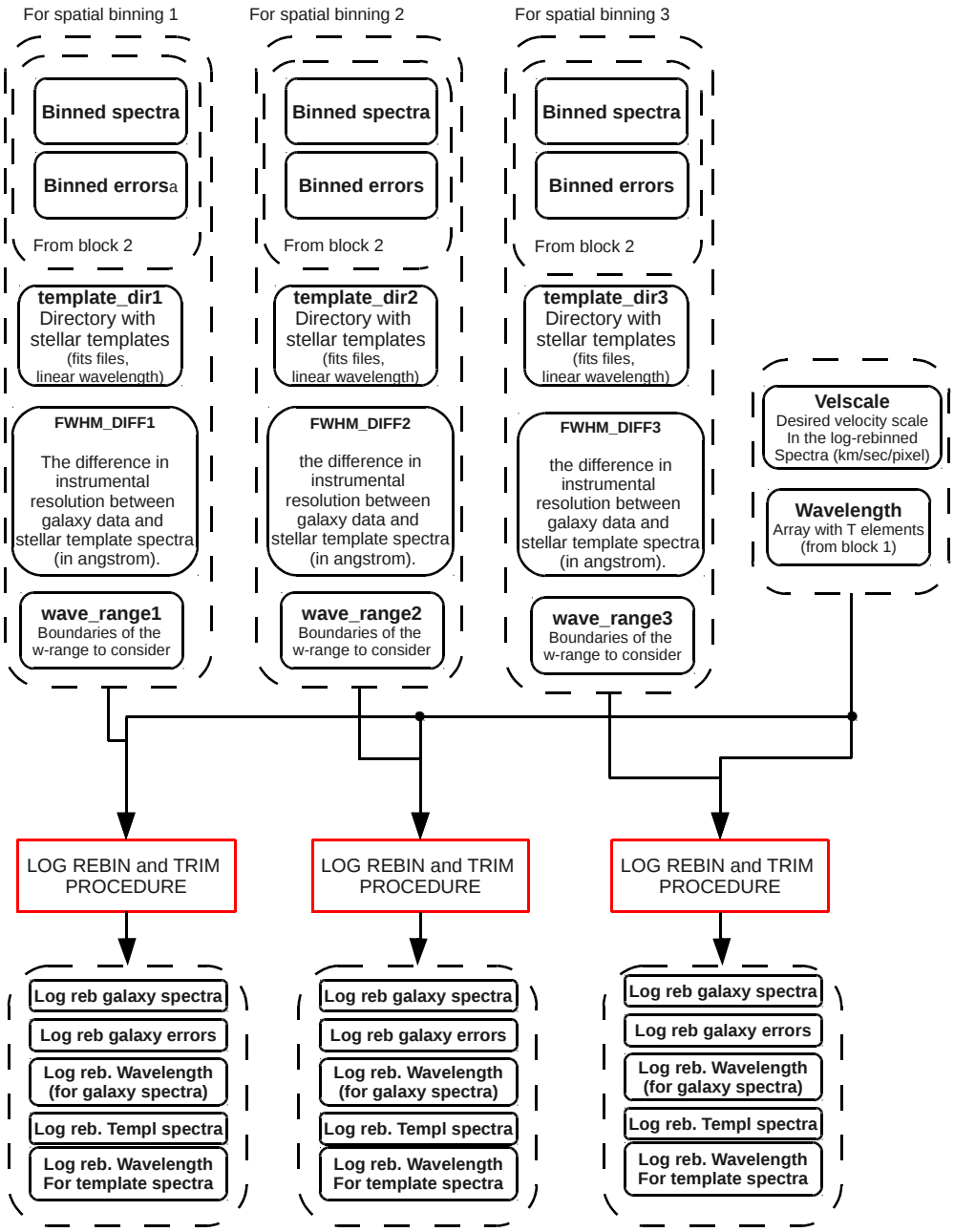

Figure 2.3: Workflow chart of the block 3 of the Data Analysis Pipeline.

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 **??**.

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	[dbl 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	[dbl array]. Vector containing the FWHM(lambda) that the stellar spectra must be convolved for. At the moment, the value median(fwhm_diff/wavelength*c) is used for broadening. Implementation to use the full LSF(lambda) information are foreseen.
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.
/gal_wavelength_log_step	Set this keyword if the input galaxy spectra are logarithmically sampled (i.e. wavelength has a logarithmic progression).
/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 3.3.1).
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 responsable 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.

The main modules in this block are mdap_spectral_fitting.pro (see Section 3.3.2), mdap_get_2d_map.pro (see Section 3.4.1), and mdap_create_starting_guesses.pro (see Section 3.3.4).

The module mdap_spectral_fitting.pro is an interface, that arranges inputs and outputs for the routine that performs the actual fit, i.e. mdap_sgandalf.pro (called by mdap_spectral_fitting.pro), which is an improved version of the pPXF routine by **?** (see Section 3.3.3 for information on how the fit is performed).

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 stellar library models (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 stellar 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). If the Milky-Way extinction value is provided, the input galaxy spectra are corrected before fitting. 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 iline 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 modules mdap_get_2d_map.pro (Section 3.4.1) and mdap_create_starting_guesses.pro (Section 3.3.4). 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 1.1.2). If the Milky-Way extinction value is provided, the input galaxy spectra are corrected before fitting. The final high level products produced in this execution are the stellar kinematics (V,σ, H3, and H4).

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 spatial sampling and *fixed* in the third execution. The spatial resampling step is performed by the modules mdap_get_2d_map.pro (Section 3.4.1) and mdap_create_starting_guesses.pro (Section 3.3.4). 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

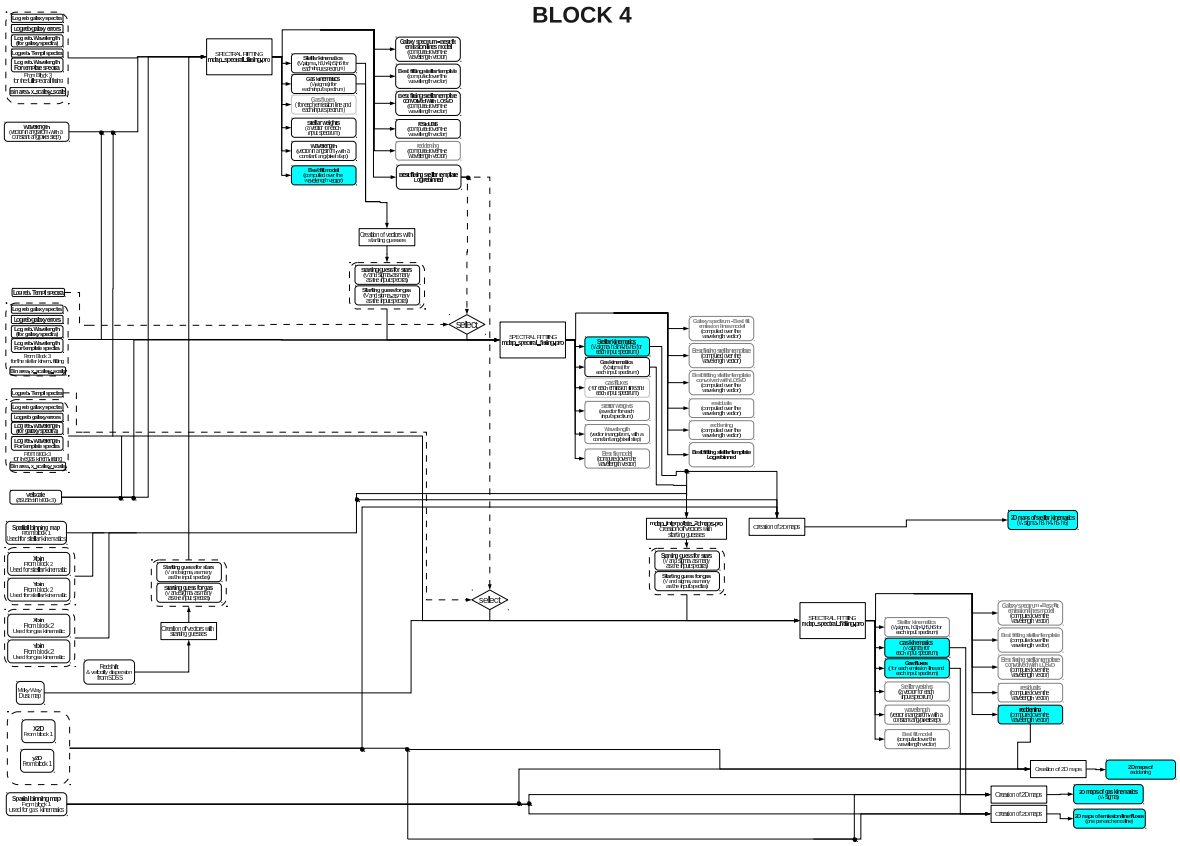

Figure 2.4: Workflow chart of the block 4 of the Data Analysis Pipeline.

keyword /dont_remove_ null_templates of the manga_dap is set, see Section 1.1.2). If the Milky-Way extinction value is provided, the input galaxy spectra are corrected before fitting. The final high level products produced in this execution are: the emission line kinematics (V,σ), reddening, and fluxes and equivalent widths (reddening corrected).

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.

At the end of block 4, the two-dimensional fields of stellar kinematics (V, σ, h3, and h4) emission line kinematics (V, σ), emission line fluxes and equivalent widths, and stellar template weights for each of the spatial binning scheme are provided. A two-dimensional map of the reddening is also computed for the first and third spatial binning scheme. Best fit spectra are also produces as output, in the rest frame wavelength or in the observed wavelength frame, according to the user's input request.

Figures 2.4 and 2.5 shows the flowchart of block 4.

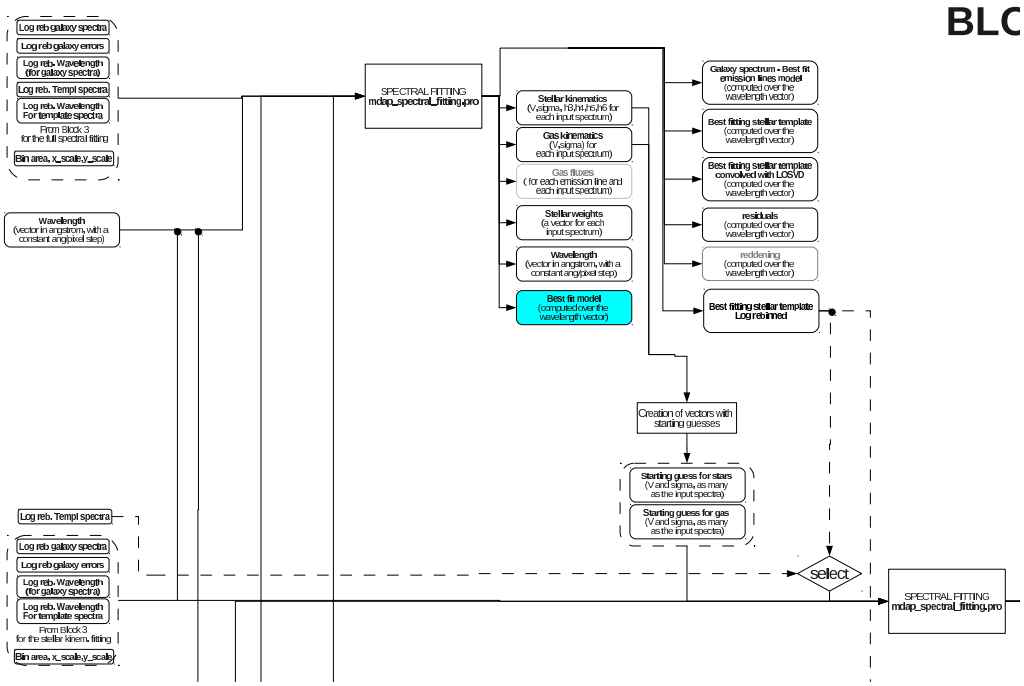

Figure 2.5: Workflow chart of the block 4 of the Data Analysis Pipeline, zoom around the first fit section.

2.5 DAP Block 5: Measurement of absorption line indices

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

It requires the following inputs:

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

Figure 2.6 illustrates the flowchart of this block.

The main module in this block is mdap_measure_ indices.pro (see Section 3.3.6), which calls the procedure that performs the measurement mdap_do_ measure_index.pro (see Section ??).

Figure 2.6 illustrates the flowchart of the main module in this block.

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

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.

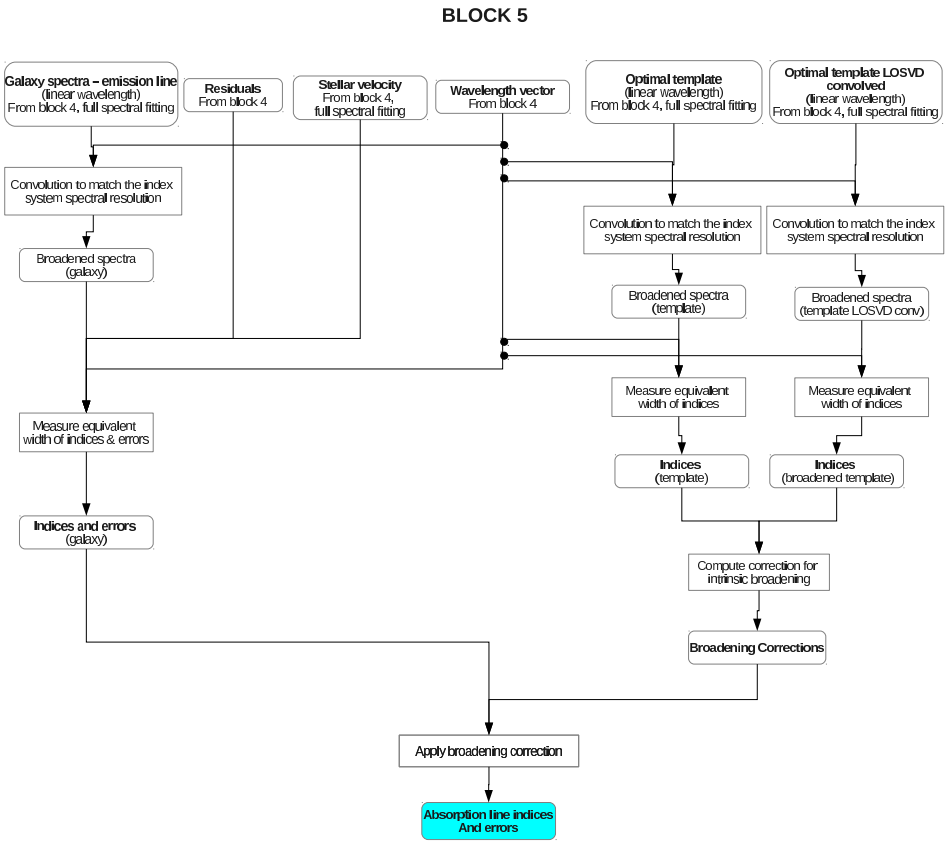

Figure 2.6: Workflow chart of mdap_measure_indices.pro, the main module in block 5 of the Data Analysis Pipeline.

Chapter 3

Main modules

In this Dection we describe the individual modules in the DAP.

3.1 mdap_calculate_spectrum_sn.pro

bla bla bla

3.2 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 3.1 lists the inputs and outputs required for this module.

Table 3.1: Inputs and outputs parameters of mdap_voronoi_2d_binning

INPUTS	
X	[flt array] Vector containing the X coordinate of the pixels to bin. Arbitrary units can be used (e.g. arcsec or pixels). In what follows the term ?pixel? refers to a given spatial element of the dataset (sometimes called ?spaxel? in the IFS community): it can be an actual pixel of a CCD image, or a spectrum position along the slit of a long-slit spectrograph or in the field of view of an IFS (e.g. a lenslet or a fiber). It is assumed here that pixels are arranged in a regular grid, so that the pixel size is a well defined quantity. The pixel grid however can contain holes (some pixels can be excluded from the binning) and can have an irregular boundary. See the above reference for an example and details.
Y	[flt array]. Vector (same size as X) containing the Y coordinate of the pixels to bin.
SIGNAL	[flt array]. Vector (same size as X) containing the signal associated with each pixel, having coordinates (X,Y). If the 'pixels' are actually the apertures of an integral-field spectrograph, then the signal can be defined as the average flux in the spectral range under study, for each aperture. If pixels are the actual pixels of the CCD in a galaxy image, the signal will be simply the counts in each pixel.
NOISE	[flt array]. Vector (same size as X) containing the corresponding noise (1 sigma error) associated with each pixel.
TARGETSN	[float]. The desired signal-to-noise ratio in the final 2D-binned data. E.g. a S/N$\sim$50 per pixel may be a reasonable value to extract stellar kinematics information from galaxy spectra.
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).'
OUTPOUTS:	
BINNUMBER	[flt array]. Vector (same size as X) containing the bin number assigned to each input pixel. The index goes from zero to Nbins-1. This vector alone is enough to make *any* subsequent computation on the binned data. Everything else is optional!
XBIN	[flt array]. Vector (size Nbins) of the X coordinates of the bin generators. These generators uniquely define the Voronoi tessellation.
YBIN	[flt array]. Vector (size Nbins) of Y coordinates of the bin generators.
XBAR	[flt array]. Vector (size Nbins) of X coordinates of the bins luminosity weighted centroids. Useful for plotting interpolated data.
YBAR	[flt array]. Vector (size Nbins) of Y coordinates of the bins luminosity weighted centroids.
SN	[flt array]. Vector (size Nbins) with the final SN of each bin.
NPIXELS	[flt array]. Vector (size Nbins) with the number of pixels of each bin.
SCALE	[flt array]. Vector (size Nbins) with the scale length of the Weighted Voronoi Tessellation, when the /WVT keyword is set. In that case SCALE is *needed* together with the coordinates XBIN and YBIN of the generators, to compute the tessellation (but one can also simply use the BINNUMBER vector).

3.3 mdap_calibrate_sn

T.B.D.

Future implementations

These implementations are foreseen:

- At the moment, the procedure broadens the input stellar spectra (log rebinned) by a Gaussian function (i.e. constant km/sec/pixel). In the future, a $LSF(\lambda)$ must be given as input to match the full spectral resolution a function of wavelength.
- Handle input galaxy spectra with a ln-lambda sampling.

3.3.1 `mdap_do_log_rebin.pro`

This procedure is called by mdap_log_rebin.pro 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.

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!

3.3.2 `mdap_spectral_fitting.pro`

This module is responsable for i) arranging the inputs from previous modules/blocks and feed them into the actual spectral fitting procedure, and ii) re-arrange the outputs of the spectral fitting procedure for further processing.

For each of the input galaxy spectra, the mdap_spectral_fitting module performs the following steps

- corrects the input spectrum by the Milky Way reddening, if provided. The correction is done using the mdap_dust_calzetti.pro provided within the GALDALF package, by M. Sarzi (see Section **??**).
- fits the de-reddened input spectrum with a set of stellar templates and Gaussian emission lines to get the absorption and emission line kinematics, the emission lines fluxes and equivalent widths, the weighs of the stellar templates, the best fit models for stars and gas, the best fitting stellar template and the best fitting stellar template convolved by the LOSVD, the residuals, and the reddening (if required).

- Resamples the output spectra over a constant λ/pixel wavelength vector.

The list of input/output parameters is given in Table 3.2.

Table 3.2: Inputs and outputs parameters of mdap_spectral_fitting.pro

INPUTS	
galaxy	[MxN dblarray]. It contains the N galaxy spectra to fit, logarithmically sampled (natural log). Units: 1e-17 erg/s/cm^2/Angstrom.
noise	[MxN dblarray]. It contains the N error vectors for the N galaxy spectra. Same units as galaxy.
loglam_gal	[M dblarray]. It contains the log wavelength values where galaxy and noise spectra are sampled.
templates	[MM x NN dblarray]. 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.
velscale	[float]. Defines the 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','BIAS=0', 'reddening=0', 'LAMBDA=exp(loglam_gal)']`
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.
MW_extinction	[float]. Milky-way extinction value at the target position (MAG). If non-zero, the input N galaxy spectra will be de-reddened using the mdap_dust_calzetti.pro routine before fitting. Default= 0 mag.

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

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, i.e. the wavelength vector to interpolate the final results on.
OPTIONAL KEYWORDS	
/use_previos_ guesses	If set, the starting guesses for spectrum $i-$th will be the best fit values from spectrum $(i-1)-$th (i> 0). Input starting guesses will be ignored.
/fix_star_kin	If set, the stellar kinematics are not fitted. The return values is that of the starting guesses.
/fix_gas_kin	If set, the emission-lines kinematics are not fitted. The return values is that of the starting guesses.
/quiet	If set, some information are not printed on screen.
/rest_ frame_log	If set, the output spectra (galaxy_minus_ ems_fit_model, best_fit_model, residuals, best_template, and best_template_LOSVD_conv) are produced at rest-frame wavelength.
OUTPUTS	
stellar_ kinematics	[N x 5 flt array]. It contains the best fit values of V, σ, h3, h4, and χ^2/DOF for each of the N fitted input galaxy spectra. If /fix_star_kin is set, the array is not defined.
stellar_ kinematics_err	[N x 5 flt arrary]. It contains the errors to the best fit values of V, σ, h3, h4, and χ^2/DOF for each of the N fitted input galaxy spectra.

stellar_weights	[N x NN dbl array]. It contains the weights of the NN templates for each of the N input galaxy spectra.
emission_line_ kinematics	[N x 2 flt array]. It contains the best fit values of V, σ (emission lines) for each of the N fitted input galaxy spectra. If /fix_gas_kin is set, the array is not defined.
emission_line_ kinematics_err	[N x 2 flt array]. It contains the errors to the best fit values of V, σ (emission lines) for each of the N fitted input galaxy spectra. If /fix_gas_kin is set, the array is not defined.
emission_line_ fluxes	[N x T flt array]. It contains the fluxes of the T fitted emission lines for each of the N input galaxy spectra. Values are corrected for reddening.
emission_line_ fluxes_err	[N x T flt array]. Errors associated to emission_line_fluxes
emission_line_ equivW	[N x T flt array]. It contains the Equivalent widths of the T fitted emission lines for each of the N input galaxy spectra. Equivalent widths are computed by the ratio of emission_line_fluxes and the median value of the stellar spectrum within 5 and 10 σ from the emission line. σ is the emission line velocity dispersion.
emission_line_ equivW_err	[N x T flt array]. Errors associated to emission_line_equivW .
wavelength_output	[QQ elements flt array]. It will contain the linear wavelength values over which the output spectra are sampled. Default: it is set to wavelength_input (if defined), or automatically computed with the smallest lambda/pixel step obtained from exp(loglam_gal).
best_fit_model	[N x QQ flt array]. It will contain the best fit models for each of the input galaxy spectra (dereddended if MW_extinction is not zero), sampled over wavelength_output. It is in rest-frame if the keyword /rest_frame_log is set.
galaxy_minus_ ems_fit_model	[N x QQ flt array]. It will contain the input galaxy spectra minus the emission lines best fit models (dereddended if MW_extinction is not zero), sampled over wavelength_output, for each of the N input spectra. It is in rest-frame if the keyword /rest_frame_log is set
best_template	[N x QQ flt array]. It will contain the best fitting template for each of the N input galaxy spectra sampled over wavelength_output (rest frame wavelength).
best_template_ LOSVD_conv	[N x QQ flt array]. It will contain the best fitting template for each of the N input galaxy spectra convolved by best fitting LOSVD and sampled over wavelength_output (rest frame wavelength).
reddening_ output	[float]. best fit value for the reddening, if the fit is required (otherwise the variable is not defined). To fit the reddening, you have to pass a starting guess value and the LAMBDA = exp(loglam_gal) vector through the extra_keyword parameter. Example: extra_inputs = ['reddening=0', 'LAMBDA=exp(loglam_gal)'].

residuals	[N x QQ flt array]. It contains the difference between the observed galaxy spectra (dereddened if the MW_reddening is defined) and the best_fit_model, sampled over wavelength_output. It is in rest-frame if the keyword /rest_frame_log is set
OPTIONAL OUTPUTS	
version	string specifying the module version. If requested, the module is not execute and only version flag is returned.

3.3.3 `mdap_sgandalf.pro`

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

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

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

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

1. an optimal stellar template is build as linear combination of stars in a library. Regularization over the stellar population properties can be used.
2. an optimal emission line spectrum is build as linear combination of N Gaussian emission lines. The number N of the emission lines and their rest-frame wavelength are read from an input file. This step should account for instrumenal LSF (to be developed).
3. the optimal stellar template is convolved with a Gauss-Hermite line of sight velocity distribution, parametrized by V, σ, h3, and h4.
4. the optimal emission line spectrum is convolved with a Gaussian line of sight velocity distribution, parametrized by V and σ.
5. the convolved optimal emission line spectrum and optimal template are multiplied by set of legendre polynomials, or by a reddening function, parametrized by $E(B-V)$ according to the mdap_dust_calzetti.pro function (Section **??**).
6. spectra from points 4 and 5 are added together and compared to the input galaxy spectrum.
7. points from 1 to 6 are repeated till minimum χ^2 is reached.

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

Table 3.3: Inputs and outputs parameters of mdap_sgandalf

INPUTS	
TEMPLATES1	[dbl array]. Vector containing the spectrum of a single template star or more ; commonly an array of dimensions TEMPLATES[nPixels,nTemplates] containing ; different templates to be optimized during the fit of the kinematics. ; nPixels has to be $\geq$ the number of galaxy pixels. Tips: • To apply linear regularization to the WEIGHTS via the keyword REGUL, TEMPLATES should be an array of two TEMPLATES[nPixels, nAge], three TEMPLATES[nPixels, nAge, nMetal] or four TEMPLATES[nPixels,nAge,nMetal,nAlpha] dimensions, depending on the number of population variables one wants to study. This can be useful to try to attach a physical meaning to the output WEIGHTS, in term of the galaxy star formation history and chmemical composition distribution. In that case the templates may represent single stellar population SSP models and should be arranged in sequence of increasing age, metallicity or alpha along the second, third or fourth dimension of the array respectively. • TEMPLATES and GALAXY do not need to span the same wavelength range. However an error will be returned by SGANDALF, if the velocity shift in pixels, required to match the galaxy with the templates, becomes larger than nPixels. In that case one has to truncate either the galaxy or the templates to make the two rest-frame spectral ranges more similar.
TEMPLATES2:	[N x 2 dbl array]. It contains: • TEMPLATES2[*,0] the values of the wavelenghts of the N emission lines to fit, in logarithmic units (ln or Log10 as in the input galaxy spectrum). • TEMPLATES2[*,1] The N signs of the gas lines to fit, +1 for emission lines, -1 for absorption lines (e.g. NaI.)

GALAXY:	[dbl array]. It contains the spectrum of the galaxy to be measured. The star and the galaxy spectra have to be logarithmically rebinned but the continuum does *not* have to be subtracted. The rebinning may be performed with the mdap_do_ log_rebin.pro (Section 3.3.1). For high redshift galaxies, one should bring the spectra close to the restframe wavelength, before doing the SGANDALF fit, to prevent too large velocity shifts of the templates. This can be done by dividing the observed wavelenghts by (1+z), where z is a rough estimate of the galaxy redshift, before the logarithmic rebinning. TO BE IMPLEMENTED IN THE PIPELINE.
NOISE:	[dbl array vector]. It contains the 1*sigma error (per pixel) in the galaxy spectrum. IMPORTANT: the penalty term of the sgandalf method is based on the *relative* change of the fit residuals. For this reason the penalty will work as expected even if no reliable estimate of the NOISE is available (see Cappellari & Emsellem [2004] for details). If no reliable noise is available this keyword can just be set to: NOISE = galaxy*0+1 ; Same weight for all pixels.
VELSCALE:	[float]. Velocity scale of the spectra in km/s per pixel. It has to be the same for both the galaxy and the template spectra.
START:	[6 elements vector]. [velStart_ stars,sigmaStart_ stars, h3, h4, velStart_ gas, sigmaStart_ gas] with the initial estimate for the velocity and the velocity dispersion in km/s.
OPTIONAL INPUTS	
BIAS:	[float]. This parameter biases the (h3,h4,...) measurements towards zero (Gaussian LOSVD) unless their inclusion significantly decreses the error in the fit. Set this to BIAS=0.0 not to bias the fit (Default value used in the DAP): the solution (including [V,σ]) will be noisier in that case. The default BIAS should provide acceptable results in most cases, but it would be safe to test it with Monte Carlo simulations. This keyword precisely corresponds to the parameter \lambda in the Cappellari & Emsellem (2004) paper. Note that the penalty depends on the *relative* change of the fit residuals, so it is insensitive to proper scaling of the NOISE vector. A nonzero BIAS can be safely used even without a reliable NOISE spectrum, or with equal weighting for all pixels.
DEGREE:	[integer]. degree of the *additive* Legendre polynomial used to correct the template continuum shape during the fit. Default: DEGREE = -1, i.e. no additive polynomial are fitted.
GOODPIXELS:	[integer array]. It contains the indices of the good pixels in the GALAXY spectrum (in increasing order). Only these pixels are included in the fit. If the /CLEAN keyword is set, in output this vector will be updated to contain the indices of the pixels that were actually used in the fit.

LAMBDA:	[dbl array]. When the keyword REDDENING is used, the user has to pass in this keyword a vector with the same dimensions of GALAXY, giving the restframe wavelength in Angstrom of every pixel in the input galaxy spectrum, i.e. lambda = EXP(logLam).
MDEGREE:	[integer]. degree of the *multiplicative* Legendre polynomial (with mean of 1) used to correct the continuum shape during the fit (default: 0). The zero degree multiplicative polynomial is always included in the fit as it corresponds to the weights assigned to the templates. Note that the computation time is longer with multiplicative polynomials than with the same number of additive polynomials. IMPORTANT: Multiplicative polynomials cannot be used when the REDDENING keyword is set.
MOMENTS:	[integer]. Order of the Gauss-Hermite moments to fit. Set this keyword to 4 to fit [h3,h4] and to 6 to fit [h3,h4,h5,h6]. Note that in all cases the G-H moments are fitted (nonlinearly) *together* with [V,sigma]. If MOMENTS=2 or MOMENTS is not set then only [V,sigma] are fitted and the other parameters are returned as zero. If MOMENTS=0 then only the templates and the continuum additive polynomials are fitted and the WEIGHTS are returned in output.
REDDENING:	[Float]. Set this keyword to an initail estimate of the reddening $E(B-V) >= 0$ to fit a positive reddening together with the kinematics and the templates. After the fit the input estimate is replaced with the best fitting $E(B-V)$ value. The fit assumes the exctinction curve of Calzetti et al. (2000, ApJ, 533, 682) but any other prescriptions could be trivially implemented by modifying the function SGANDALF_REDDENING_CURVE within the procedure. IMPORTANT: The MDEGREE keyword cannot be used when REDDENING is set.

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

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

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

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

3.3.4 mdap_create_starting_guesses.pro

This module takes the kinematic output of mdap_spectral_fitting.pro, which defined over an irregular spatial grid, and interpolates them over another irregular spatial grid. The procedure handles the stellar kinematic output (V, σ, h3, and h4, in this case the keyword \h3h4 must be set) as well as the emission line kinematic output (V, σ). It is used in block 4 to map a set of measurements obtained over a specific spatial binning into a new spatial binning, so that they can be used as starting guesses for a future binning. The interpolation is actually performed by the mdap_interpolate_2dmaps.pro routine (see Section 3.3.5), which uses the GRID_TPS IDL function.

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

Table 3.4: Inputs and outputs parameters of mdap_create_starting_guesses.pro

INPUTS	
input_map	[N x I flt arrary]. Array of values measured on the N galaxy spectra. The value of I must be either 2 (in the case the measurements refer to the emission line kinematics, V and σ), or 4 (in the case the measurement refers to the stellar kinematics.) If I=4, the keyword \h3h4 must be set.
input_xbin	[N elements array]. X coordinates of the N spatial bins (in arcsec), were the input_map is defined.
input_ybin	[N elements array]. Y coordinates of the N spatial bins (in arcsec), were the input_map is defined.

x2d	[NNxMM flt array]. X coordinates of the field of view, in arcsec (produced in block 1 by mdap_read_datacube.pro). Coordinate 0 is the center of the field of view.
y2d	[NNxMM flt array]. Y coordinates of the field of view, in arcsec (produced in block 1 by mdap_read_datacube.pro). Coordinate 0 is the center of the field of view.
velocity_initial_guess	[float]. Value to be used for the velocity initial guess, in the case there are no enough points for spatial interpolation ($N < 2$).
velocity_dispersion_initial_guess	[float]. Value to be used for the velocity dispersion initial guess, in the case there are no enough points for spatial interpolation ($N < 2$).
H3_initial_guess	[float]. Value to be used for the H3 initial guess, in the case there are no enough points for spatial interpolation ($N < 2$).
H4_initial_guess	[float]. Value to be used for the H4 initial guess, in the case there are no enough points for spatial interpolation ($N < 2$).
output_xbin	[M elements array]. X coordinates of the M spatial bins (in arcsec), were the input_map needs to be interpolated (e.g. produced by mdap_spatial_bin.pro).
output_ybin	[M elements array]. Y coordinates of the M spatial bins (in arcsec), were the input_map needs to be interpolated (e.g. produced by mdap_spatial_bin.pro).
OPTIONAL KEYWORDS	
/h3h4	If set, allows for the creation H3 and H4 starting guesses. Default, only V and σ are computed.
OUTPUTS	
output_start_guess	[M x I dbl array]. Quantities in the input_map interpolated over the coordinates output_xbin and output_ybin. If \h3h4 is set, $I = 4$, otherwhise $I = 2$.

3.3.5 `mdap_interpolate_2dmaps.pro`

This procedure is called by mdap_create_starting_guesses.pro (Section 3.3.4) and interpolates a set of measurements, which defined on a irregularly sampled spatial grid, over another irregularly sampled spatial grid. It uses the GRID_TPS internal function to perform the interpolation.

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

Table 3.5: Inputs and outputs parameters of mdap_interpolate_2dmaps.pro

INPUTS	
input	[N elements double array]. The N values to interpolate (defined over the x,y coordinates).

x	[N elements flt array]. X coordinates in arcsec corresponding to the N values (it is the same vector as input_xbin in mdap_create_starting_guesses.pro).
y	[N elements flt array]. Y coordinates in arcsec corresponding to the N values (it is the same vector as input_ybin in mdap_create_starting_guesses.pro).
x2d_full	[NNxMM flt array]. X coordinates of the field of view, in arcsec (produced in block 1 by mdap_read_datacube.pro). Coordinate 0 is the center of the field of view. It is the same array as x2d in mdap_create_starting_guesses.pro.
y2d_full	[NNxMM flt array]. Y coordinates of the field of view, in arcsec (produced in block 1 by mdap_read_datacube.pro). Coordinate 0 is the center of the field of view. It is the same array as y2d in mdap_create_starting_guesses.pro.
x_out	[M elements flt array]. X coordinates in arcsec over which input must be interpolated (it is the same vector as output_xbin in mdap_create_starting_guesses.pro).
y_out	[M elements flt array]. Y coordinates in arcsec over which input must be interpolated (it is the same vector as output_ybin in mdap_create_starting_guesses.pro).
OUTPUTS	
output	[M elements double array]. input values interpolated over the x_out, y_out coordinates.

3.3.6 mdap_measure_indices.pro

This module is responsible for measuring the strength of the absoprtion line indices (performed with mdap_do_measure_indices.pro, see Section 3.3.8), their errors, and correct them for galaxy intrinsic broadening. Broadening correction is applied to the input galaxy spectra to match the spectral resolution of the spectral indices system (e.g. Lick). Input galaxy spectra must have emission lines removed.

Broadening correction is done according to the following formula

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

for atomic indices, and

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

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

The list of input/output parameters for this module is given in Table 3.6.

Table 3.6: Inputs and outputs parameters of mdap_measure_indices.pro

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

3.3.7 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.

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.

3.3.8 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 3.8.

Table 3.8: Inputs parameters and return value of mdap_dust_calzetti.pro function

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

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

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

3.4.1 `mdap_get_2d_map.pro`

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

Table 3.9 lists the inputs and outputs required for this module.

Table 3.9: Inputs and outputs parameters of mdap_get_2d_map.pro

INPUTS	
values	[Nbins elements dbl array]. Measured quantity for each spatial bin (first elements is associated with bin 0 , second element with bin 1, and soforth). It can be either an output of block 4 or block 5.
binning	[N x M dbl array]. Two dimensional map showing the binning sheme. Pixels beloning to the i+1-th bin have value i (i= 0, 1, ..., Nbins-1). Pixels associated to no spatial bin have value -1 (produced in block 2 by mdap_spatial_binning.pro). Nbins is the number of spatial bins.

x2d	[N x M flt array]. X coordinates of the field of view, in arcsec (produced in block 1 by mdap_read_datacube.pro). Coordinate 0 is the center of the field of view.
y2d	[N x M flt array]. Y coordinates of the field of view, in arcsec (produced in block 1 by mdap_read_datacube.pro). Coordinate 0 is the center of the field of view.
OUTPOUTS	
map	[N x M flt array]. Measured values over the two-dimensional field of view. Pixels belonging to the same spatial bin have the same value (i.e. no spatial smoothing or interpolation are performed).

3.4.2 `mdap_display_2dmap_ps.pro`

T.B.D.