Siena3D User Manual

N. Winkel[1] and M. Singha[2]

[1]Max-Planck-Institut für Astronomie, Königstuhl 17, D-69117 Heidelberg, Germany, winkel@mpia.de
[2]Department of Physics & Astronomy, University of Manitoba, Winnipeg, MB R3T 2N2, Canada

October 24, 2022

Abstract

SIENA3D is a tool to analyse the morphology of the extended emission line region morphology in 3D spectroscopic observations of AGN. It performs a multi-Gaussian fit of the AGN emission line spectrum and traces different kinematic components across the field of view by keeping the kinematic parameters fixed. In addition, SIENA3D models the empirical PSF extracted from the data to measure the projected size, location and luminosity of the different kinematic components. SIENA3D is developed based on unobscured (type 1) AGN that were observed with ESO VLT/MUSE. The technique can in principle be applied to 3D cubes obtained with any other IFU instrument.

1 Installation

1.1 Software dependencies

SIENA3D depends on the following software:

- python version 3.10 or later
- numpy, version 1.12.1 or later
- scipy, version 0.19.0 or later
- astropy, version 2.X or later
- maoppy, version 1.3.1

While different combination of package version may work as well, we suggest to create a dedicated python environment with miniconda[1]. All packages can be easily install within the conda environment using the command

```
conda install PACKAGE=VERSION
```

which allows the user to select the necessary specific versions described above to be installed under a common environment. This procedure ensures stable operations of SIENA3D . Running SIENA3D with different version than the once listed above is at the users risk.

1.2 Package installation

We recommended to create a dedicated conda environment prior to the installation:

```
conda create --name Siena3D
conda activate Siena3D
```

In order to install SIENA3D , one can obtain the package from PyPi with

```
pip install Siena3D
```

If the user does not want to create a dedicate environment with conda, you need to consider running the command with superuser-permissions or to change the command to either of the following commands

[1] https://docs.conda.io/en/latest/miniconda.html

```
python install . --user
python install . --home
```

while the preferred option is clearly the first, the second option requires a correct setup of the `PYTHONPATH` environment variable.

In addition, the user needs to manually install the Maoppy[2] package with the commnand

```
git+https://gitlab.lam.fr/lam-grd-public/maoppy.git@master
```

1.3 Example files

The Siena3D package comes with several example files in the form of 3D-spectroscopic data cubes. They demonstrate the setup for running Siena3D and help to get the user familiar with the necessary parameter files, input files and output files.

Important: The installation procedure will **not** copy the example files to a new location, but will stay in the examples directory. The user is free to move these files to any location. However, the location of the input files must be specified in the parameter files and is independent of the internal Siena3D installation directory.

2 Quick start

This section gives an overview of how to set up and execute Siena3D quickly. It provides the minimum information to run Siena3D successfully on a certain data set. It is based on the example files that come with the Siena3D package. We recommend that new users first run Siena3D on the example files and have a careful look at each input files, parameter files and output files. After that before applying Siena3D to their own data. More details on the algorithms and full explanations on all steps of the workflow are given in Sect. 3. Sect. 5 provides information on how to set up the input files. For the most common errors and difficulties we refer to the trouble shooting Sect. 8.

2.1 The input and parameter files

2.1.1 Required input files

Below we briefly explain the necessary files to run Siena3D and are‘ available in the example directory:

- `Input/MockCube.fits` → Input spectral data to fit
- `parameters.par` → Parameter file for setting up the input data
- `eline.par` → Definition of the emission line multi-component model
- `incl.cont` → Wavelength selection for the continuum fit
- `excl.fit` → Wavelength mask for the emission line fitting

Please check that all those files are available in the example directory of the downloaded package.

2.1.2 Setting up the parameter files

To successfully run Siena3D a correct setup of the necessary parameters are crucial. Most of the parameters may only be changed by expert users and can be left safely unchanged for most circumstances. We refer to Sect. 3 to gain an insight into the underlying algorithms and purpose of all parameters that can be set to optimise Siena3D .

2.2 Running Siena3D

Siena3D must be applied to a 3-dimensional cube from integral field spectroscopy. If the user directs to the example directory, the command below will execute the procedure.

```
python runall.py
```

This will start the workflow for one of the example files included in the package.

[2] https://gitlab.lam.fr/lam-grd-public/maoppy

2.3 The output files

Below we briefly sumarise the output files that SIENA3D produces during running.

- `<PREFIX>.minicube.fits` → truncates the input data cube
- `<PREFIX>.AGN.html` → best-fit AGN spectrum and the residual spectrum in an interactive HTML file
- `<PREFIX>.AGNspec.fits` → best-fit AGN spectrum and the residual spectrum in a FITS file
- `<PREFIX>.best_model_components.fits` → AGN spectrum, error and the spectra of the individual components
- `<PREFIX>.par_table.fits` → parameters of the emission line components
- `maps/<component>.fits` maps of the kinematic components' 2D light distribution
- `Spectroastrometry.jpg` → summary plot with an example spectrum and the spectroastrometry maps

Please check if all files are available in the `Output` directory after running SIENA3D on the the example data set.

3 Methodology

In this section the methodology is described by which SIENA3D identifies and traces the 2D morphology of different kinematic structures in the AGN spectrum across a 3D data cube.

3.1 Overview

A schematic of the individual steps that SIENA3D uses is sketched in Fig. 1. The user provides SIENA3D with an input 3D datacube which is first pre-processed and from which the AGN spectrum is extracted. After fitting the AGN spectrum (see Sect. 3.3), the kinematic 'base' spectra are set up according to the input parameter files specified by the user (see Sect. 3.4. The kinematic components are then fitted to the entire data cube (see Sect. 3.5). The resulting 2D light distribution is then fitted with the PSF to constrain the size of the analysed structure (see Sect. 3.6).

3.2 Pre-processing

Depending on the IFU instrument with which the data was acquired, the input data cube first needs to be cropped to (*i*) a smaller wavelength and (*ii*) smaller field of view (FOV). This has two reasons. First of all, the emission lines on which the diagnostics will be performed usually lie in a relatively narrow wavelength range, e.g. the Hβ-[O III] window or the Hα-[N II] window in the optical. To avoid contamination from additional features and large-scale continuum variations, it is sufficient to crop the wavelength axis to the respective windows. The second reason is that each of the spaxles will be fit with the kinematic component spectra (see Sect. 3.5) which is computationally expensive if carried out for the entire data cube. Since the ionized gas structures for which SIENA3D was developed are typically close to the nucleus and small in size (i.e. barely resolved), SIENA3D first crops the FOV to a square window around the AGN location that specified by the user in the `parameters.par` file (see Sect. 5.1) Also the spatial size of the window can also be specified in the parameter file.

3.3 Fitting the AGN spectrum

From the pre-processed input data cube, SIENA3D extracts the AGN spaxel from the position that the user specifies in the `parameters.par` file (see Sect. 5.1) or automatically extracts the highest-S/N spectrum in the minicube.

In the next step, SIENA3D attempts to reproduce the AGN spectrum with a multi-component emission line model. The individual emission line components and their initial guess parameters must be specified by the user in the `eline.par` file (see Sect.5.2). In addition to the listed emission line components, SIENA3D uses a simple power-law to model the AGN continuum.

Important: The power-law is a valid assumption only if (1) the fitted wavelength range (2) the AGN continuum dominates over the stellar continuum emission.

The details of the minimisation routine are described in the following subsection Sect. 3.3.1. As a result of this process, SIENA3D produces an interactive HTML figure (see Fig. 2) that pops up after the best-fit model has been found. We encourage the user to vary the initial guess parameters to validate that the best-fit solution produces robust results.

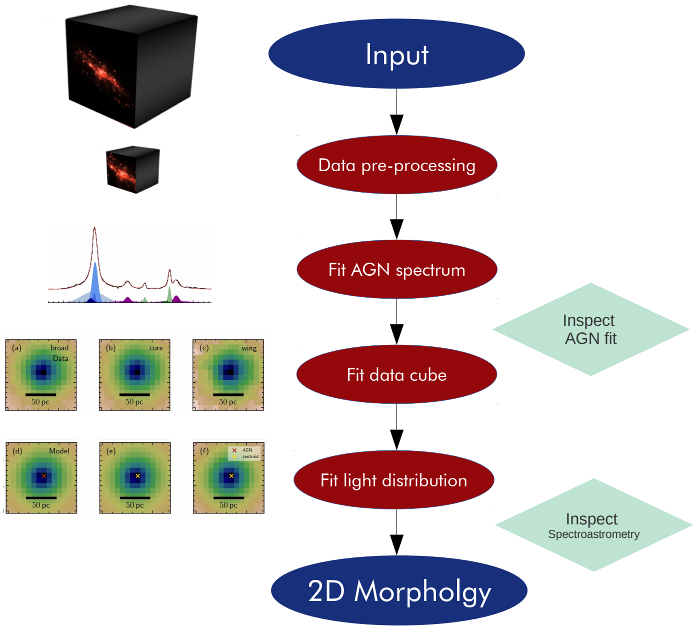

Figure 1: Schematic of the SIENA3D methodology. The workflow is structured in 5 consecutive steps. (1) Read in and pre-process the input data to set up the 3D window (the ’minicube’) on which the analysis will be performed. (2) Fit the AGN spectrum to identify the kinematic components which will be spatially traced (3) Fit the minicube to spatially map the contribution from the kinematic components. (4) Fit the 3D light distribution with the PSF model to locate the components relative to the AGN position. (5) Constrain the kinematic components’ morphology from the residual structure. After steps (2) and (4) respectively, there are interactive plots showing up in which the user can validate the result of the previous step.

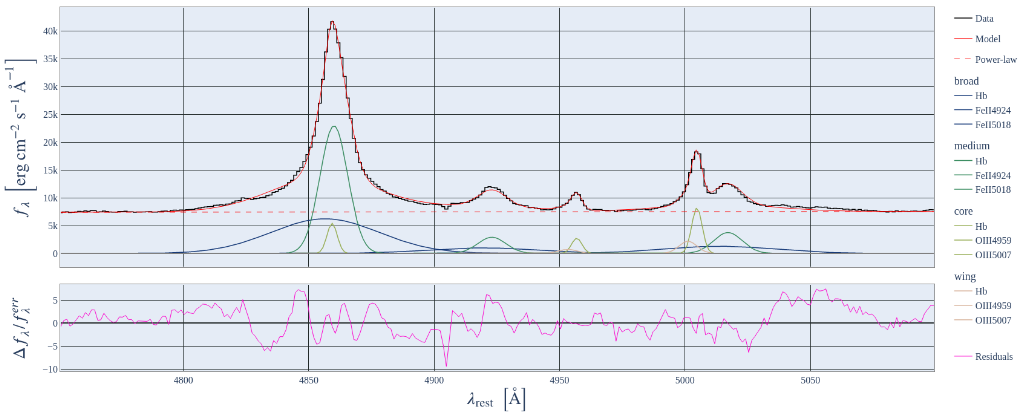

Figure 2: SIENA3D Example for the result of the AGN spectrum. This interactive html plot contains the AGN spectrum together with the best-fit model in the top panel. Furthermore, the contributions of the continuum and the emission line components are shown. The cursor can be moved over the lines to show to which component the line belongs. The bottom panel shows the normalised residuals between AGN spectrum and the best-fit model.

3.3.1 Emission-line modelling

Fitting emission lines is a non-linear process. We employ standard χ^2 minimisation algorithms included in the astropy.modeling[3] package to find the best-fit parameter set for a given model. In a spectrum of a galaxy usually several emission lines of interest are present so that a typical model is usually of a superposition of several lines across the spectrum. In Siena3D we use the widely adopted Levenberg-Marqudart algorithm to minimise the χ^2 of the model.

The issue with fast χ^2 algorithms is that they require a good starting point for exploring the χ^2 surface to find the global minimum and not being trapped in a local minimum. Therefore, reasonable starting values are necessary are crucial for successful non-linear emission-line modelling.

> **Important:** Since Siena3D uses the best-fit model its components across the cube, it is crucial that the AGN spectrum robustly reproduced by the multi-component emission line model. The user might want to adjust the number of components and initial guess parameters in the `eline.par` file to find the most robust solution.

3.3.2 Line profiles

Only simple **Gaussian** line profiles are available in the current version of Siena3D . We plan to expand this further to Gauss-Hermite expansions of lines to take asymmetric line shapes into account. If asymmetric lines are present in the data, currently two separate Gaussian profiles need to be used to capture the asymmetry of lines implicitly. This can be realised by adding another (kinematically tied) component to the the emission line in the `eline.par` file.

Gaussian line

The Gaussian line profile is described by the central wavelength, line dispersion, and an integrated line flux. In PyParadise the central wavelength of a line is computed from the rest-frame wavelengths specified in the `src/siena3d/data/eline_rf.txt` file together with a velocity that is the redshift expressed in velocity space as $v\,[\mathrm{km/s}] = c \times z$. The line dispersion is expressed by an *intrinsic* velocity dispersion. It is used to compute the line dispersion at a given redshift which is then enlarged in quadrature by the provided instrumental spectral resolution in units of [Å]. Hence, an emission line is parameterised by four parameters which need to be provided for each Gaussian component in the model. Of the four parameters, only three free parameters are optimised.

A key feature of Siena3D is that coupling redshift, line dispersion and fixing certain line ratios is straight forward. The user can simply indicate which components belong to the same component which can significantly reduce the number of free parameters and makes the optimisation process more robust. Fixing line ratios is meant for known multiplets of lines with a-prior priors on their ratios.

3.4 Generating the kinematic component basis

After having the the best-fit kinematic parameters of the emission line components in the AGN spectrum determined, Siena3D initialises a set of basis spectra which combine the individual emission line spectra among the different components. For each kinematic component specified in the `eline.par` file, Siena3D combines the emission lines in which this component is present by keeping the kinematic parameters as well as the relative contribution between the emission lines fixed. In this way we generate a set of base spectra, i.e. the kinematic components' spectra, which may have have tied kinematics. The resulting base spectra are then normalised to a an integrated flux of 1 in units of the input data cube.

3.5 Fitting the cube

In order to trace the kinematic components spatially, Siena3D fits the base spectra to each of the spaxels across the minicube. Since each of the components must have a non-negative contribution and the optimisation is a linear problem, we use a non-negative least squares (NNLS) algorithm to find the best-fit solution. More precisely, Siena3D uses the scipy.optimize.nnls of the scipy[4] package which solves

$$arg\,min||Ax-b||^2 \ \text{ for x} > 0. \qquad (1)$$

Here, b represents the spectrum that we want to fit, A represents the matrix that contains the base spectra, x is a vector that contains the coefficients x_i that are multiplied to the base spectra, and $||.||_2$ denotes the Euclidean norm. In this notation,

[3] https://www.astropy.org/

[4] `https://scipy.org/`

the coefficient x_i represents the contribution of the component i the normalised base spectra to the input spectrum b. Thus, mapping the best-fit coefficients x_i across the FOV leaves us with the 2D light distribution from the component i.

3.6 Modelling the PSF

If the kinematic components originate from a region close to the galaxy nucleus that is hardly resolved, its spatial offset and its intrinsic extent are difficult to identify by eye. In particular the finite width of the PSF complicates the measurement. In order to determine the exact location of the 2D light ditribution, and to measure the intrinsic extent of the emission SIENA3D fits a PSF to the 2D light distribution of each of the kinematic components. The different options from which the user can select the analytic model are listed in Table 1. For seeing-limited observations the Moffat profile is an appropriate description. For the adaptive-optics assisted narrow field mode (NFM-AO) of the MUSE instrument, (Fétick et al., 2019) have provided an analytic model which is included in SIENA3D .

File name	application	type	Reference
PSFAO19	adaptive optics shaped PSF	analytic	Fetick+19
Moffat	seeing-limited	analytic	
Gaussian	core of seeing-limited PSF	analytic	

Table 1: Available PSF models in the current SIENA3D distribution.

By construction, the broad emission line component in spectra of unobscured AGN is spatially unresolved. The PSF model should therefore provide a good description of the broad component's 2D light distribution. If the residuals show significant and extended substructure, the kinematic components are degenerate or the selected PSF model is not an accurate description. In many cases, the systematic uncertainty from the PSF model are larger than the statistical error in the observational data. SIENA3D uses distribution of the residual fluxes to estimate a systematic uncertainty. The systematic error is also taken into account when the uncertainties of the spectroastrometric measurement are determined with the Monte Carle approach (see Sect.4).

3.7 Spectroastrometry

Once the best-fit PSF models have been fit to the components' 2D light distribution have been determind, SIENA3D computes the offset centroid location w.r.t. the centroid of the BLR emission. This offset corresponds to the line-of-sight projected distance from the AGN location. Fig. 3 shows the final figure which SIENA3D produces during this last step. It contains am example spectrum from one spaxel in the minicube, together the 2D light distribution of the selected components specified in the `parameters.par` file. The user can inspect each spectrum of the minicube by clicking any pixel in the maps on the right.

3.8 Emission line region morphology

A routine to estimate the size of the emitting structure is currently being developed but not included in the current distribution.

4 Monte Carlo error estimation

The importance of error estimation

The spectral fitting only provides the mean values for the emission line parameters and no errors. The Monte Carlo method is technique to obtain estimates for the uncertainties on the emission line parameters and their location. The `samples_eline` in the `parameters.par` file gives the number of Monte Carlo samples that should be performed during the emission line fitting. The `samples_astrometry` in the `parameters.par` file gives the number of Monte Carlo samples that should be generated during the spectroastrometry. SIENA3D already obtained the errors on the emission line kinematics when SIENA3D fits the spectra for the first time. As for every Monte Carlo sample a new datacube is generated, the error estimation is performed relatively slow, especially if the analysed window is big.

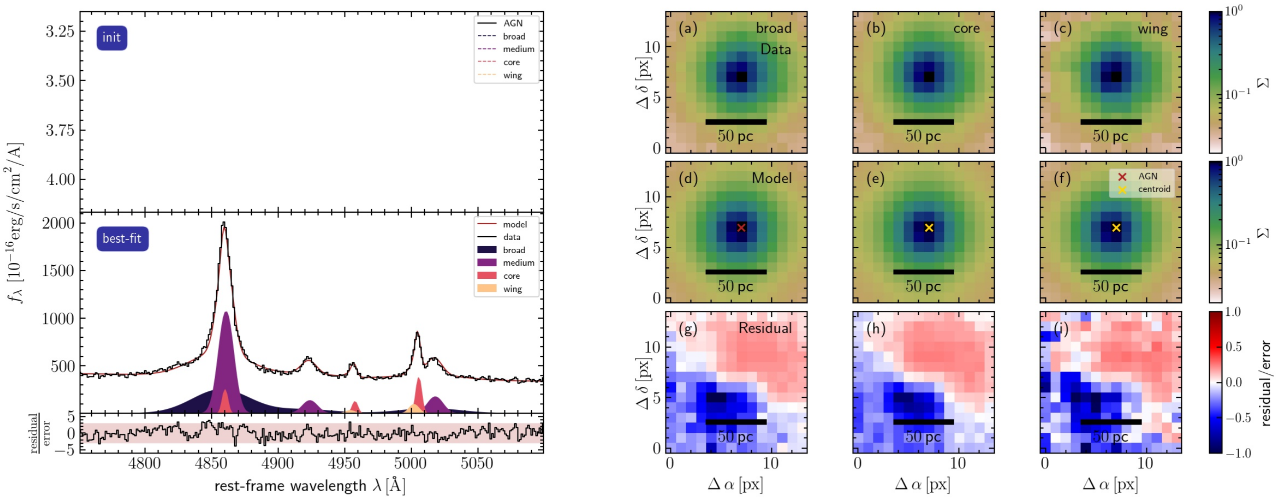

Figure 3: SIENA3D results for the first spectrum in the provided example file `MockCube.fits`. The top right panel shows the AGN spectrum together with the setup of the kinematic basis. The bottom left panel shows the best fit AGN spectrum together with the contribution of the best-fit model which consists of the kinematic components. The panels on the right hand side show (top) the empirical 2D light distribution from fitting the components to the cube (middle) the best-fit PSF model to the empirical map (bottom) the residual maps, normalised by the uncertainty.

4.1 Errors - Emission line fitting

In each of the steps of the SIENA3D workflow where we fit spectra with an emission line model, we estimate the errors with a Monte Carlo approach. From the error spectrum, we generate a given number of artificial spectra for which we repeat the fitting. The best-fit parameters are determined from the median and half of the 16^{th} to 84^{th} percentile respectively. The number of MC can be adjusted by the user (see 5.1)

4.2 Errors - Spectroastrometry

For the spectroastrometry, we generate a given number of artificial datacubes, for which we repeat the entire process described in Sect. 3.5 to Sect. 3.6. Again, we adopt the median and half of the 16^{th} to 84^{th} percentile for each of the parameters best-fit and uncertainty respectively. The number of MC samples can be adjusted by the user (see 5.1)

5 Setting up the input files

5.1 parameters.par - Parameters file

The parameter file `parameters.par` is an ASCII file which sets several important parameters and organises the workflow. Each line defines a certain parameter with a parameter name, a parameter value, and optionally followed by a comment separated with a '#' sign from the parameter. It contains information the location of the input files, the region in the input datacube that will be analysed as well as important parameters that may be adjusted by the user to improve the spectroastrometric analysis of the input data. A typical setup of the file looks like this:

```
obj          test                 # object name
input        Input/Mrk1044.fits   # input data cube
cz           4913.4               # systemic velocity of the source
instr_fwhm   2.54                 # instrumentals resolution [Angstroem]
wvl_start    4750                 # lower wavelength limit for emission line fitting
wvl_end      5100                 # upper wavelength limit for emission line fitting
sampling     0.025                # spatial sampling of the input data cube [arcsec/px]
ncrop        14                   # size of the patch around the QSO location which will be analysed [px]
coor         2,2                  # coordinates for which spectrum in final plot will be shown
psf_model    PSFAO19              # model for the PSF, choose from `Gauss`, `Moffat`, `PSFAO19`
plotmaps     broad,core,wing      # kinematic components that shall be included in the final plot
```

The most important parameters to be updated by the user for every data set are `obj`, `input`, `cz`, `wvl_start` and `wvl_end`. The absolute or relative path to the input data cube containing the stellar continuum template library `input` need to be set properly. It is crucial to set the wavelengths window parameters such that the included window contains all the emission lines together with regions where the AGN continuum dominates. If the user uses different instrument than VLT/MUSE NFM-AO also the parameters `instr_fwhm`, `psf_model` need to be updated. All other parameters can be left unchanged except the user is familiar with the details of parameters, and knows the consequences.

The parameters are the the following:

- `obj` (`string`): object name
- `input` (`string`): path of the input data cube, relative from the working directory.
- `cz` (`float`): systemic velocity of the galaxy. This value corresponds to a redshift expressed in velocity space as $v_{\text{sys}} = c \times z$.
- `instr_fwhm` (`float`): instrumentals resolution of the input data cube in units of [Å]
- `wvl_start` (`float`): lower wavelength limit for emission line fitting.
- `wvl_end` (`float`): upper wavelength limit for emission line fitting. excl.fit incl.cont
- `incl_cont` (`string`): Path to the file which contains the wavelength windows to exclude during the determination of the stellar continuum. This may be an absolute path or a relative path.
- `excl_fit` (`float`): Path to the file which contains the wavelength windows to exclude during the emission line fit. This may be an absolute path or a relative path.
- `sampling` (`float`): spatial sampling of the input data cube in units ["/px].
- `ncrop` (`int`): size of the patch around the AGN location which will be analysed [px].
- `coor` (`tuple`): spatial spaxel coordinates of the cropped cube for which spectrum in the final plot will be shown. Tuple values must be integers.
- `psf_model` (`string`) model for the 2D PSF in the image plane. Available options are ‘Gauss‘, ‘Moffat‘, ‘PSFAO19‘.
- `plotmaps` (`string`): kinematic components that will be included in the final plot. The value needs to consist of a subset of the components listed in the
- `samples` (`int`): The total number of random walks performed inside an MCMC chain. The longer the chain, the more reliable becomes the determination of v and σ.

5.2 elines.par - emission line components

Siena3D will fit emission line profiles to the individual spectra after the best-fit power-law continuum model is subtracted. Another parameter file is needed for this second processing step which has a different structure as the input parameter file. The `eline.par` file is a ASCII file that defines a list of emission line components that will be included in the model with which the emission line spectra will be fitted. In this file the user can specify how many components for each emission line will be used and how the kinematic parameters are tied among the lines. Furthermore, the user must specify an initial guess for each of the components. Each kinematic component consists of a block of several lines depending on the specific component as described below. **A typical parameter file looks like this**:

```
# eline             component   tied        amp_init   offset_init   stddev_init
Hb_broad            broad       Hb_broad    .15        0             22
FeII4924_broad      broad       Hb_broad    .05        0             22
FeII5018_broad      broad       Hb_broad    .05        0             22
Hb_medium           medium      Hb_medium   .3         0             5
FeII4924_medium     medium      Hb_medium   .1         0             5
FeII5018_medium     medium      Hb_medium   .1         0             5
Hb_core             core        Hb_core     .2         0             2
OIII4959_core       core        Hb_core     .2         0             2
OIII5007_core       core        Hb_core     .2         0             2
Hb_wing             wing        Hb_wing     .1         0             5
OIII4959_wing       wing        Hb_wing     .2         0             5
OIII5007_wing       wing        Hb_wing     .2         0             5
```

From top to bottom the table lists the individual emission line components which will be combined to reproduce one line shape. The structure for each component is the same. From left to right the columns contain the following information that characterise the respective components:

- `eline` (string): Emission line to which the component belongs
- `component` (string): Kinematic component to which the emission line component belongs.
- `tied` (string): Component to which the kinematics are be tied. Emission line components with the same 'tied' value will have the same kinematic parameters during both the AGN spectrum fit and the spectroastrometric analysis. If a component with free kinematic parameters is included in the model, its 'tied' value can be set to either the component's name or 'None'.
- `amp_init` (float): initial guess for the strength of the component, relative to the strongest emission line in the analysed wavelength-window.
- `offset_init` (float): initial guess for the offset of the central wavelength with respect to the systemic velocity. Value given in units of [Å].
- `stddev_init` (float): initial guess for the line dispersion. Value given in units of [Å]

An example parameter file `eline.par` is available in the `example` directory. However, the included parameters usually need to be updated for every data set by the user. If the components are at rest relative to the galaxy frame, the velocity offset `offset_init` can be set to zero. For outflowing ('wing') components, however, we recommend to specify the expected line shift with `stddev_init`, since. In general, we recommend to try fitting AGN spectrum with as little components as possible. Only if the single component model is insufficient to reproduce the shape of individual or multiple lines additional components should be added. Then, it is crucial to correctly couple the components' kinematics among each other since the fitting result will be highly degenerate instead. Since free parameters are usually updated automatically based on the empirical data, it is recommended that the brightest line is set with free parameters (i.e. set `tied` to the `eline` value or 'None') in case coupling between several lines is desired. This will provide the best guess value for coupled parameters.

Important: The key to a successful run of SIENA3D and to reduce computational time is a careful look at the `eline.par` file. It is crucial that fitting the AGN spectrum produces robust results. In some cases, the relative contribution from the individual components might vary dramatically, depending on the initial parameters and kinematic coupling of the emission lines. We strongly encourage the user to test different setups before interpreting the results produced in the next step. More details on setting up the parameter files are described in Sect. 5.1.

5.3 Wavelength selection files

SIENA3D requires two different wavelength selections for which dedicated files exist. SIENA3D requires three wavelength selection files for different purposes: 1) exclusion of emission lines, sky line residuals and detector artefacts during the continuum normalization → `excl_cont` in `parameter_stellar`, 2) exclusion of non-continuum emission for the actual stellar continuum fit → `excl_fit` for `parameter_stellar`, and 3) wavelength windows covering ALL emission lines to be fit → `lines.fit` for `parameter_eline`. For most practical purposes `excl_cont` and `excl_fit` use the same spectral windows and can point to the same file for simplicity.

Important: Failure in providing correct and complete wavelength selection files will usually lead to incorrect fitting of the spectra. This will be recognised as certain artefacts in the results. .

incl.cont - Selection for the continuum model

As the spectra are first continuum-subtracted before AGN spectrum and the individual spaxels are fitted, one needs to select regions which are suitable for the continuum modelling, hereafter referred as continuum selection. It is crucial to select regions which are free from prominent emission and absorptions lines and artefacts/residuals, strong emission lines and strong absorption lines. We note that SIENA3D was developed to analyse unobscured AGN host galaxies in narrow window near the nucleus. There, the spectra are dominated by beam-smeared AGN emission and two narrow windows at the blue and red end respectively are sufficient to model the prominent AGN-continuum with a power-law.
For the simple case of fitting the Hβ-[OIII] window, the wavelength range file would look like:

```
# [rest-frame]
4750 4755
5080 5090
```

excl.fit - Mask for the emission line model

The second set of wavelength ranges that needs to be specified involves the regions in the spectrum that must not be included during the emission line modelling. This can be artificial structures in the spectra of regions which are dominated by strong absorption lines.

Usually the emission lines are not spread over the full wavelength range of the input spectra. Computation speed and robustness is gained if the modelling of the emission lines is restricted to the wavelength range the emission line actually contribute significantly to. Therefore, SIENA3D only considers the wavelength range specified in the `parameters.par` file. it is highly recommended to provide a file that defines certain wavelength intervals for which the model is actually evluated and a χ^2 value is computed. The syntax of this file is the same as for the wavelength masking for the stellar continuum fitting, but only wavelength ranges defined in the rest-frame will be used. For the simple case of fitting the [OIII] double as defined in the Model file example above the wavelength range file would simply look like:

```
# [rest-frame]
4930 5030
```

This means that only the wavelength range from 4930 to 5030 (in Angstrom) would be fitted and the rest of the spectrum would be ignored. Of course, the unit of the wavelength ranges needs to match with that of the spectrum and adjusted as necessary.

6 Data formats

Like any software package PyParadise also requires that data is provided in certain pre-defined formats to run successfully. Failure of the program may often be related to invalid input. Therefore, users are strongly encourage to read this section carefully before using this software for their analysis.

6.1 Spectral data

Siena3D can can only be feed with one basic type of 3D-spectroscopic data. The minimum information it must contain is the wavelength grid and the corresponding flux densities that makes up the spectra. Optimal but often important information are the associated errors and known bad pixels. The data structure must follow the specified format to be recognised by Siena3D .

General structure of spectral FITS files

There is no standard for storing spectral information in FITS files. The format used by Siena3D is described in the section, where a general description of FITS files is given and the specific information to the 3D cube files are given in Sect. 6.1.1.

A FITS file consists of several HDUs, Header Data Units. The spectra (`DATA`), the errors (`ERROR`), and the pixel mask (`BADPIX`) are stored in different HDUs. The `DATA` HDU should be the first HDU and include a few header values regarding the wavelength information. Any other information is not ignored (`DC-FLAG` still needs to be made consistent). In the header of `ERROR`, the header keyword `EXTNAME` should be set to `ERROR`. Similarly, in the header of `BADPIX`, the header keyword `EXTNAME` should be set to `BADPIX`. No other header information will be read in from `ERROR` and `BADPIX`.

6.1.1 Data cubes

A cube is the primary output of an integral field unit (IFU), and can also be the end-products of radio telescope observations. It is a combination of an image, with at every spatial position (referred as spaxel) a spectrum. The stored data should have the structure of $x \times y \times z$ pixels, where x and y span the spatial space and z is the number of wavelength elements. The wavelength information of a cube is given by the header keywords `CRVAL3`, first wavelength element, and `CDELT3`, the wavelength difference between the i^{th} and $(i+1)^{\text{th}}$) pixel, in units of Ångstrom per pixel. As any standard and valid FITS file, it should contain the header keyword `NAXIS3`, which equals z, and it should contain the header keyword `NAXIS1` and `NAXIS2`, which equal to x and y respectively.

7 Output files

Pre-processing input data

As a first step, a folder in the working directory named `Output` is created where all the output files will be written to. Siena3D then truncates the input data cube to the size specified in the `parameters.par file` and stores it in `<PREFIX>.minicube.fits`.

AGN spectrum fit

The best-fit AGN spectrum and the residual spectrum are stored as an interactive HTML file and `<PREFIX>.AGN.html`. An example for the resulting stellar continuum model for the test case `Mrk 1044` is shown in Fig. 3. Furthermore, a binary FITS table `<PREFIX>.AGNspec.fits` with the same dimension as in the input data cube along the spectral axis is stored.

A FITS table `<PREFIX>.best_model_components.fits` is stored contains the AGN spectrum, its error and the spectra of the individual components fitted to the AGN spectrum. In addition, a FITS binary table `<PREFIX>.par_table.fits` is stored which contains the parameters of the emission line components including errors from the MCMC run.

Spectroastrometry

Furthermore, for each of the different components specified in `eline.par`, a 2-dimensional map of the light distribution is stored in the sub-diretory `maps` as `<component>.fits` The figure that shows an example spectrum fit together with the maps of the kinematic components is stored in `Spectroastrometry.jpg`.

8 Troubleshooting

9 Frequently asked questions

The AGN spectrum is poorly fitted? If the fitted AGN spectrum does not match the observed one,[...]

Acknowledgements

We thank Bernd Husemann who contributed to this project and with whom we developed the concept behind SIENA3D .

References

R. J. L. Fétick, T. Fusco, B. Neichel, L. M. Mugnier, O. Beltramo-Martin, A. Bonnefois, C. Petit, J. Milli, J. Vernet, S. Oberti, and R. Bacon. Physics-based model of the adaptive-optics-corrected point spread function. Applications to the SPHERE/ZIMPOL and MUSE instruments. *Astronomy & Astrophysics*, 628:A99, August 2019. doi: 10.1051/0004-6361/201935830.