EXPLORE S-Phot and PySSED manual
(EXPLORE published version, PySSED version 0.3)

Iain McDonald

September 2023

Contents

1 Introduction

EXPLORE is a programme funded by the European Union, promising six scientific data applications (**SDAs**) to allow innovative scientific data exploration and exploitation applications for space sciences. See: `https://explore-platform.eu/`

- **S-Phot** is one of these SDAs, providing an application to explore stellar photometry from photometric surveys of the sky, specifically including the *Gaia* satellite data.
  - **PySSED** (Python Stellar Spectral Energy Distributions) is a Python 3 programme that underlies S-Phot.

The purpose of this document is to explain how to use the S-Phot SDA and PySSED software. It does not describe the mechanisms behind the software's workings, or the physics that underlies the programming choices. Those are dealt with in an associated publication in RASTI.

PySSED has two versions. One can be run remotely within EXPLORE's S-Phot SDA. This is most useful for small and simple runs, and benefits from a graphical user interface (GUI) to simplify interaction. The second is a command-line version, designed to be installed locally, and is most suitable for power-users who are likely to require significant computing power or who wish to make adjustments to the detail of how PySSED works.

2 Remote access through EXPLORE

PySSED is installed as part of the S-Phot scientific data application (SDA) on the EXPLORE platform (`https://explore-platform.eu/`), where outputs can be visualised and interacted with. Full details on the additional functionality can be found in the User Manual associated with the SDA.

3 Local installation

PySSED is available from as a single ZIP file for use in any system running Python 3:
`ttps://gitub.com/iain-mcdonald/PySSED/` `ttps://gitlab.acri-cwa.fr/project-explore/sda-spot` To install PySSED, simply create a directory, move the ZIP file into that directory, and unzip it.

To avoid taking up large volumes of space, the full stellar atmosphere models are not included in the ZIP file. This means that it is not possible to include additional filters in the reduction. Details on how to obtain this functionality are listed later in this section.

3.1 Prerequisites

PySSED requires a Python 3 installation with the following packages:
`sys numpy scipy astropy pandas itertools matplotlib mpl_toolkits wget astroquery`.

The `dust_extinction` packages is required if you want to do any extinction correction. Depending on the extinction correction you wish to do, you will either need the `gtomo` package from EXPLORE, which is included in the ZIP file, or the `dust_maps` package.

The `datetime` and `time` packages are strongly recommended but their absence should not prevent the programme from running except in some cases of poor server connections, though information on the programme's wall time usage will be unavailable.

3.2 Stellar atmosphere models and adding new filters

For speed and space reasons, PySSED refers to a pre-computed table of stellar atmosphere models, detailing the expected flux for a star of specific properties at a standard distance in each of the allowed filters (see filters input file). This is ok for most use, except if you want to change the available filters from the default settings. Removing filters from the computation does not affect this file, but adding new filters requires this file to be recomputed.

This pre-computed file is ~20 MB, compared to the original models, which are ~85 GB. You will therefore need to make sure you have both the free space to download *and process* these data, and the few days needed to reduce it to a sensible size. It is best to run these as a series of overnight tasks.

3.2.1 Getting the models

Stellar atmosphere models are freely available from many locations. However, for inter-operability, PySSED relies on the Spanish Virtual Observatory format. Unfortunately, there exists no means of downloading an entire set of models from the SVO, so the means for downloading these models must become "creative". The process here describes the means to download PySSED's standard model set (BT-Settl AGSS2009 — other model sets are available, but aren't currently supported). It is important to ensure that each step has completed successfully before moving on to the next step.

First, navigate to the `models/` directory. This should be empty apart from three files: `wget.bash`, `reduce.bash` and `reduce2.bash`.

Running "`source wget.bash`" will download the BT-Settl AGSS2009 models, which are the default models used by PySSED (note you will need `wget` installed to do this). This will take some time (~1 day!) and should give you a complete listing of files from `1.asc` to `14196.asc`. Once complete:

- Check that all files have downloaded by running "`ls | wc -l`", which should result in 14 199 files, including the three `.bash` scripts.
- Check that every file has downloaded in full by running "`ls -l | awk '$5<1e6'`". This will highlight files less than 1 MB in length. The only files that should appear are the `.bash` scripts.

If any files have not downloaded correctly, you can re-run the specific lines of `wget.bash` again.

The models then need renamed into a sensible format. Running "`reduce.bash`" will take all the `*.asc` files and rename them according to their header information into `t****g+*.**m+*.**a+*.**` format (where `t` is temperature in Kelvin, `g` is $\log(g)$ in cgs units, and `m` and `a` are [Fe/H] and [α/Fe], respectively). This also reduces the data file to a single wavelength and flux measurement.

This reduction process will take some time (~1 day) due to the size of the data. If it is interrupted, run the "`reduce2.bash`" file to run the remainder of the files.

3.2.2 Reducing the models to a lookup table

Begin by navigating to be `src/` directory. At this point, it is a good idea to make a backup, so that you can roll back your changes:

```
mkdir backup
cp model-bt-settl.dat backup/
cp model-bt-settl-recast.dat backup/
```

Now we need to add the filters to the filter input file. It's best to create a new filter input file to do this:

- `cp filters.default filters.new`
- Edit `setup.default` and change `FilterFile` from `filters.default` to `filters.new`.
- Edit `filters.new` and add the filters you want to add to the computation (see section on input files for details).
- To make use of these new filters, you will also need to edit the catalogues input file so that the programme knows which catalogue tables to look up for photometry in this filter.

Now we need to reduce the model atmospheres to a simpler look-up table.
python3 makemodel.py bt-settl
This will again take 1–2 days! It will result in a new file `model-bt-settl.dat`.

3.2.3 Completing the model grid

The model atmosphere look-up table is an incomplete grid. For faster on-the-fly multi-dimensional interpolation, it is important that this grid is completed. However, this is yet another process that will take 1–2 days!

First, edit `setup.default` (or whichever setup file you want to use). Change `RecomputeModelGrid` to 1. *(It is very improtant you change this back again afterwards to avoid doing the computation again unnecessarily!)* While you are there, check that the parameters `Model*Lo` and `Model*Hi` cover the ranges of $T_{\rm eff}$, $\log(g)$, [Fe/H] and [α/Fe] you need. These should be appropriate for the BT-Settl models.

The model grid can then be completed by running any PySSED command, e.g.:
`python3 pyssed.py single "Betelgeuse" simple` As part of this process, PySSED will download any filter information it needs from the SVO and recast the look-up table into a complete look-up grid named `model-bt-settl-recast.dat`.

Once this is done, you can optionally run the `shorten-model.scr` script to reduce the size of the recast model grid, which has the benefit of a slight speed improvement.

Don't forget to change `RecomputeModelGrid` back to zero, otherwise, that should be it!

4 PySSED inputs

4.1 Running PySSED

PySSED can be run directly from the command line, or as a module within a Python environment. To run PySSED from the command line, use:
`python3 pyssed.py <cmdtype> [cmdparams] <proctype> [setup file]`
To run PySSED from within Python, use:
`pyssed(cmdtype, cmdparams, proctype, procparams, setupfile)`
where the input parameters are:

- `cmdtype`: The type of search you wish PySSED to perform. This can be a `single` object, a `list` of objects or an area (`cone`, `rectangle` or `box`) search.
- `cmdparams`: The object(s) or area you wish PySSED to query.
  - For a single object, this should be the name of the object in quotes, e.g. `"EU Del"`. This name will be queried at SIMBAD. Alternatively, a *Gaia* source number can be queried directly, e.g. `"Gaia DR3 1812371096773414144"`, or a set of sexagesimal co-ordinates given, e.g. `"20 37 54.728 +18 16 06.895"`.
  - For a list of objects, this should be the filename containing a set of single objects, in the same formats as above. This should again be in quotes, e.g. `"objects.list"`.
  - For a cone search, this should be the co-ordinates of the cone centre and radius in decimal degrees, e.g. `"226.417 -29.008 0.1"` for a cone with $6'$ radius around Sgr A*.
  - For a rectangle search, this should be the co-ordinates of the rectangle centre, width and height in decimal degrees, e.g. `"226.417 -29.008 0.1 0.1"` for a rectangle of $6 \times 6'$ around Sgr A*.
  - For a box search, this should be the co-ordinates of the box centre, width and height in decimal degrees, e.g. `"226.361 -29.108 226.473 -28.908"` for a box of $6 \times 6'$ around Sgr A*.
- `proctype`: The type of post-processing you wish to be performed on the compiled SED. This could be:
  - `none`;
  - `bb`, which fits a blackbody; or
  - `simple`, which fits a stellar model atmosphere.
- `procparams`: This is reserved for future use by different `proctype` options. It should currently be passed as an empty array (`procparams=[]`).
- `setupfile`: Optionally, the name of the setup file that controls user options for PySSED. If not given, it defaults to `setup.default`. Details of this and the other input files used by PySSED are given below.

4.2 Input file formats

All PySSED input files are tab-separated ASCII files with a fixed number of comments. Any lines beginning with '#' are treated as comments (thus any unwanted lines can be commented out by adding a '#').

4.3 `setup` input file

The setup input file, by default `setup.default`, controls the main functionality of the PySSED programme. It also contains the user options.

The contents of the setup file are discussed primarily in:

- Section ?? for options that discuss PySSED functionality and control;
- Section ?? for options that discuss PySSED plotting and output files.

However, four of the options control the files from which PySSED takes detailed inputs. These are `SurveyFile`, `FilterFile`, `AncillaryFile` and `RejectFile`, which set the files to adopt for survey catalogues, filter properties, ancillary data and data-rejection criteria. The details and formats of these input files are given below.

4.4 `catalogues` input file

The catalogues input file, by default `catalogues.default`, defines the photometric catalogues to be queried in the fitting. (Catalogues for ancillary data are queried more generically in the ancillary input file.) Each line of the file represents a specific catalogue, with the following columns:

- `Server`: This denotes the origin of the file. Typically, the catalogue will be a table hosted on `VizieR`, although it may a local `File`.
  - VizieR interaction is generally best, though local files can be faster for very small datasets, if a specific subset of a survey is desired, or if the user has data not present on VizieR.
  - User files should begin with a comment ('#') line listing the field headers. At minimum, these should include an object name, an RA and Dec in decimal degrees, and at least one magnitude or flux measurement.
- `Name`: This is the internal name that PySSED will use to refer to this catalogue. You will need to specify this in the filters input file later, and it will appear in the output plots and data files.
- `TableID`: This is the name of the VizieR table from which to take data. For files, this is the filename relative to the `src/` directory.
- `ObjCol`: This is the column name in the table corresponding to the object name.
- `Epoch`: This is the epoch for which RA and Dec are defined in the catalogue. The equinox of the catalogues should always be J2000.0, but the epoch is typically either the year the catalogue data was taken (this is normal for many modern, deep photometric catalogues) or the year 2000.0 (this is normal for catalogues where objects are referred to by name, where VizieR refers back to their SIMBAD co-ordinates).
- `BeamSize`: This is the typical beam size of the catalogue. This may be defined by the resolution of the telescope or the typical seeing disc. It is used in area searches to define whether observations have a one-to-one match or are blended.
- `XMatchCone`: This is the cut-off radius for cross-matching in `single`-source or `list` searches (`BeamSize` is used in area searches for better treatment of blending). It normally represents the astrometric accuracy of the catalogue. The number should normally be conservatively large to allow for position uncertainties due to photometric noise (typically a fraction of `BeamSize`), global and local astrometric offsets in the catalogue, and errors caused by the fact that the catalogue `Epoch` often extends over months or years and that proper motion moves stars during this time (see control parameters `PosErrMult, AddPMErr` and `EpochPMErrYears`).

4.5 `filters` input file

The filters input file, by default `filters.default`, defines the filters used in the defined catalogues. Each line of the file represents a specific filter in a specific catalogue, with the following columns:

- `CatName`: Catalogue name, as defined in the `catalogues` input file.
- `CDSLabel`: Column label of the magnitude or flux assigned in the VizieR table.
- `CDSError`: Column label of the error in the magnitude or flux assigned in the VizieR table.
- `SVOName`: Filter name assigned by the Spanish Virtual Observatory.
- `DataType`: Defining the units used for magnitude (`mag`) or flux. Flux units can be `nJy`, `uJy`, `mJy` or `Jy` for units of Jansky and its SI-like divisions, or `nMgy` for "nano-maggies" (multiples of 10^{-9} Vega, e.g., as used by the unWISE catalogue).
- `DataRef`: When magnitudes are used, this specifies whether to use the **Vega** or **AB** magnitude zero point. When flux is used, this column must be present, but is ignored.
- `ErrType`: Type of error quoted in the catalogue. This can either be the `Same` as the flux unit (e.g., magnitudes or Jy), it can be `None` where no error exists, or it can be `Perc` if the error is quoted as a percentage.
- `MinData, MaxData`: These two columns specify the minimum and maximum magnitudes or fluxes that are accepted for SED fitting (otherwise PySSED sets `mask = False`).
- `MaxPErr`: The maximum percentage flux error accepted for SED fitting (otherwise PySSED sets `mask = False`). This also works when the input type is magnitudes.
- `ZptCorr`: Normally zero, but an offset (in magnitudes) can be applied here to arbitrarily shift the photometric zero point up and down.

4.6 `ancillary` input file

The ancillary input file, by default `ancillary.default`, defines the ancillary data the programme collects as part of the SED fitting. This can be arbitrary data, but also includes data the programme uses, such as distances. Each line of the file represents a specific piece of data to find, with the following columns:

- `Server`: The server to query for the ancillary data. This is normally `Vizier` but can be `Gaia` (this tends to be a slower query than querying the *Gaia* catalogue at VizieR. File options are not currently supported.
- `Name`: An internal name assigned to the catalogue by PySSED. If this is a catalogue that already exists in the catalogues input file, the same name should be given here.
- `TableID`: The name given to the table on the remote server (e.g., the VizieR table number).
- `ObjID`: the server's column label for the object name.
- `ColumnID`: the server's name for the column being queried. For VizieR, note that this is sometimes different from the column name given during an http query, especially when the column name includes non-alphanumeric characters.
- `UncertaintyID`: the server's name for the column containing the uncertainty on `ColumnID`. Normally this is one column denoting a symmetric error, but sometimes an upper and lower limit is given. In this case, the columns can be given as `<upper_label>/<lower_label>`. However, note that these will be averaged to give one column in the final output!
- `ParameterID`: PySSED's internal name for the parameter in question. This is used to group measurements of the same parameter together to assign `priority`. Note that there are some *special parameters* used by PySSED, namely: `RA`, `Dec`, `PMRA`, `PMDec`, `Parallax` and `Distance` (distance and parallax are treated as a single parameter). By default, these parameters will be taken from the master catalogue (normally *Gaia* or, failing that, *Hipparcos*). These defaults will be overwritten if `priority < 10`.
- `Units`: A text label describing the units of the observation. These should be identical for all observations of the same `ParameterID` and must follow conventions for the special parameters (position in decimal degrees, proper motion in $\mathrm{mas\,yr^{-1}}$, parallax in mas and distance in pc).
- `Multiplier`: A multiplicative factor applied to the data to change it to the standard `Units` (e.g., 365.25 to covert years to days).
- `Epoch, BeamSize, XMatchCone`: The epoch of observation for proper-motion correction, beam size for blending characterisation and cross-match search cone (see the catalogues input file for details).
- `Priority`: The priority used for combining data of the same `ParameterID`. A lower number indicates a higher priority: if two observations of the same parameter exist, and one is assigned higher priority, it will be chosen as the primary output in the `output.dat` file; all results will be shown in the `.anc` file, but lower-priority observations will be set to `mask = False`.

  If two or more observations exist with the same priority, then the following procedure occurs:

  - If the ratio of fractional errors is more disparate than `AncWeightingLimit`, or if the difference between the observations exceeds their quadrature-combined uncertainty by a factor greater than `AncSigmaLimit` (both variables in the setup file), then the observation with the smaller error will be used.
  - Otherwise, if the fields are numerical, they will be combined by weighted average, where the wights are defined by the inverse square of their fractional errors (if no error is given, the fraction error is set to unity).
  - If the fields are text, the first reported result will be taken.

4.7 `reject`-reasons input file

The reject-reasons input file, by default `reject.default`, defines the reasons why data should be rejected by PySSED. This may include instances where specific data is preferred over other data, or where flags denote poor-quality data. Each line of the file represents a specific reason to reject data. These can either be used to reject data in the same survey, or a different survey, by slightly different uses of the same columns:

- `CatName`: Internal name of the catalogue containing the rejection criterion.
- `CDSLabel`: Label used by CDS to define the column to be acted on (see notes on column entries above). This can be `All` if all data that survey contains about an object are to be rejected (e.g., when a "bad source" flag is raised).

- `Column`: When rejecting data based on criteria internal to a survey, this is the CDS column name that contains the rejection criterion (e.g., quality flagging data). When rejecting data based on criteria that compare two surveys, this is the type of criterion you want to query (i.e., `mag` to query magnitudes, `flux` to query fluxes, `magerr` or `ferr` to compare their respective errors, or `anc` to compare any other arbitrary column). Although it may seem convoluted, this allows a catalogue containing a flux to be compared to one containing a magnitude easily.

- `Position`: For comparison within a survey 1-indexed numerical position in the `Column` to be queried. This is useful for quality-flag columns (e.g., in 2MASS, the third column of the `Qflg` column contains the quality flag for the $K_{\rm s}$ filter). If the entire column is to be queried, or for comparison between surveys, `-1` must be entered.

- `Logical`: A logical symbol (<, > or =) that denotes the comparison to be made. For comparisons between surveys, this can be prefixed with one of four characters:
  - `d`, to denotes the difference between the catalogues' values,
  - `D`, to denote an absolute difference
  - `f`, to denote a flux ratio between the two catalogues' values, or
  - `F`, to denote an absolute flux ratio.

- `Value`: Limiting value assigned to the criterion.

- `Result`: This can be either `Mask`, in which case the data is masked but still displayed, or `Remove`, in which case the data is not displayed in the output.

- `RejSurvey`: For comparisons between surveys, the catalogue name that is being compared to. For internal comparisons, this should be `Same`.

- `RejFilter`: For comparisons between surveys, the CDS column label of the data that is being compared to. For internal comparisons, this should be `Same`.

While powerful, these rejection criteria can appear quite complicated. Hence, some examples are given below:

- `2MASS Kmag Qflg 3 = D Mask Same Same`
  This would mask the 2MASS $K_{\rm s}$ photometric data point if the third entry in `Qflg` was `D`.

- `IRAS Fnu_12 q_Fnu_12 -1 = 1 Remove Same Same`
  This would remove the *IRAS* [12] photometric data point if `q_Fnu_12` was `1` (this is a string comparison, so will not match `1.0`).

- `SDSS7 gmag mag -1 d< -1.0 Mask Hipparcos Hpmag`
  This would mask the SDSS7 g' photometric data point if it was more than a magnitude fainter than the *Hipparcos* $H_{\rm p}$ magnitude.

- `DIRBE F1.25 flux -1 F> 2 Mask 2MASS Jmag`
  This would mask the *COBE/DIRBE* [1.25] photometric data point if it was more than a factor of two brighter or fainter than the 2MASS J-band flux.

- `DIRBE F12 flux -1 F> 1 Mask IRAS Fnu_12`
  This would mask the *COBE/DIRBE* [12] photometric data point if *IRAS* [12] photometry exists.

- `WISE W1mag mag -1 > 5 Mask catWISE W1mproPM`
  If the *WISE* [3.4] magnitude is brighter than 5th magnitude, use cat*WISE* in preference.

5 PySSED controls

Controls for input and output data are described in the preceding and next sections, respectively. The master setup file (normally `setup.default`) also controls the functionality of the programme. This section steps through the sections in this file and describes the use of these controls.

- `verbosity:` This controls the amount of text that PySSED reports back to the user. For normal operations, this would be somewhere in the region of 30–60. Values approaching 99 will output a lot of text, which can be helpful if piped to an output file for debugging.

- `GaiaEpoch:` This sets the epoch of the *Gaia* catalogue, which is used as a master catalogue for proper-motion calculations. This should be set to the native epoch described in the *Gaia* data release being used.

5.1 Treatment of photometry

- `SurveyFile, FilterFile, RejectFile`: Files used for describing input surveys, filters and reasons to reject photometry. See Section 4.
- `DefaultError`: Some catalogues do not provide errors on their photometry. This value sets the default error to be assumed if flux errors are missing. The error returned will be the flux multiplied by this value.
- **Merging criteria:** When two or more observations are taken in the same filter, either the most-precise value is taken, or the observations will be merged by taking a weighted average of the fluxes. The weighting used is the inverse square of their fractional errors. The following criteria control these actions.
  - `MinPhotError`: The minimum fractional error to be assigned to photometry when merging data in the same filter, which avoids giving apparently precise photometry undue weight. This over-rides the photometry in the original data. This is an important consideration in surveys which have under-estimated photometric errors, which can occur if the uncertainty in the photometric zero-point and filter transmission curve is not included in the errors. It also allows flexibility for errors not accounted for in instantaneous photometric surveys, such as different amounts of flux from blended stars compared to other surveys, different amounts of incorporated background light, and intrinsic stellar variability.
  - `PhotWeightingLimit`: This determines the error limit used when deciding to use the most-precise photometry or to merge photometry. For a set of observations of a star, ordered from the most-precise to the least-precise, observation i, with photometric error δ_i, will be included in the merged calculation if:

    $$\frac{\mathrm{Max}(\delta_1, \texttt{MinPhotError})}{\mathrm{Max}(\delta_i, \texttt{MinPhotError})} > \texttt{PhotWeightLimit}, \tag{1}$$

    unless `PhotSigmaLimit` is not also satisfied.
  - `PhotSigmaLimit`: This determines the discrepancy limit used when deciding to use the most-precise photometry or to merge photometry. A flux m is first calculated, representing the mean flux when two observations are made in the same filter, or the median flux when three or more observations are made. For each observation, i, with flux F_i and error δ_i, the deviation of the point is calculated as

    $$\sigma_i = \frac{|F_i - m|}{\mathrm{Max}(\delta_i,\ F_i\,\texttt{MinPhotError})}. \tag{2}$$

    If $\sigma_i >$ `PhotSigmaLimit` then the most-precise photometry will be used, otherwise the photometry will be merged (provided `PhotWeightingLimit` is also satisfied).
- `VizierRowLimit`: The maximum number of rows to extract from any VizieR table. This is used to avoid unreasonably large server queries that would take unreasonably long times to process. This should be the bigger than the largest number of objects you expect to extract in one go.
- `ServerRetries`: The number of times to retry querying a server before determining that the server is unresponsive. The wait time increases for each retry. For short queries, this should be kept small. For large queries, this can be increased to deal with short-term interruptions in server operations or network traffic.
- `WarnMissingCol`: When active, this flag triggers a warning message if a catalogue query returns no data. This can be useful to diagnose when a catalogue entry is misformatted, but can be undesirable if absent data is frequent in a query.

5.2 Treatment of area searches

Terminology: A *catalogue* represents a large-area survey containing many *objects*. Objects appearing across many catalogues can be described as astronomical *sources*.

- `PrimaryCatRef`: Selected from the catalogues input file, this is the catalogue used as a primary reference for identifying sources. Objects in this primary catalogue will be converted into sources, and objects from other catalogues will be matched to them. Objects in the output file will be named on the basis of this catalogue. The catalogue should contain:
  - Proper-motion information, to correct for proper motion when performing source matching.
  - Distance information, to compute luminosities. This can also be supplied separately as ancillary information, or a fixed value can be assumed.

  Consequently, `PrimaryCatRef` should normally the latest *Gaia* data release, except for specialist observations (e.g., *HST/JWST* observations of external galaxies where distances and proper motions can be taken as a fixed value). Functionality for non-*Gaia* catalogues is *not* fully tested in this version!

- `TrimBox`: When a box search is performed, a square is extracted by VizieR, centred between the two RA and Dec extremes. If this is false, that square is preserved. If this is true, the square is trimmed into a wedge, ensuring that only sources between the two RA extremes are included. This accounts for the tapering of lines of RA towards the poles, useful for tiling surveys.
- `ReportSecondarySources`: Not supported in current version.
- `UseCatNameAsOutputName`: Not supported in current version.
- **Controls for proper-motion correction:**
  - `PMCorrType`: This determines the type of proper-motion correction that is used for identifying matches. `PMCorrType = 0` means no proper-motion correction is applied; `1` uses the proper motion from the primary catalogue (`PrimaryCatRef`) if it is available, otherwise use a fixed proper motion; `2` uses a fixed proper-motion for all stars.
  - `FixedPMRA, FixedPMDec`: This sets the fixed proper-motion to be assumed if a proper-motion correction is not available or if `PMCorrType = 2`. Values are in mas yr^{-1}.
  - `PMRAColID, PMDecColID, PMRAErrColID, PMDecColID`: If `PrimaryCatRef` is *Gaia* or *Hipparcos*, then the columns for proper motions and their errors will be automatically selected. If `PrimaryCatRef` is a non-standard catalogue, then proper motions can be selected here.
- **Treatment of blending for potential matches:** Matches between catalogues can be defined by the number of potential matches that exist within a certain radius (the catalogue's `BeamSize`). For example, when comparing a low-resolution catalogue to a high-resolution catalogue, an object in the low-resolution catalogue may have zero, one or many (more than one) objects from the high-resolution catalogue within its `BeamSize`. These leads to the following nine scenarios when finding matches to a master catalogue within a secondary catalogue:
  - *Zero/Zero:* A secondary catalogue object has no matches in the primary catalgoue. No match, but potential to report as a secondary source (`ReportSecondarySources`).
  - *Zero/One:* A secondary catalogue object has no match to any primary source within the primary catalogue's `BeamSize`, but nevertheless matches a primary source within its own (larger) `BeamSize`.
  - *Zero/Many:* A secondary catalogue object has no match to a primary source within the primary catalogue's `BeamSize`, but matches several primary sources within its own (larger) `BeamSize` (e.g., an *IRAS* detection may not lie directly on top of a *Gaia* source, but several *Gaia* sources may exist with the *IRAS* beam and be potential counterparts).
  - *One/Zero:* A primary source has no match in secondary catalogue.
  - *One/One:* A primary source has a unique match in a secondary catalogue.
  - *One/Many:* A primary source has many matches in a secondary catalogue. The primary source is identified as a blend.
  - *Many/One:* More than one object from the primary catalogue matches the same object in the secondary catalogue. The secondary object is identified as a blend.
  - *Many/Many:* The primary source has multiple matches in the secondary catalogue *and* one or more of the corresponding objects in the secondary catalogue matches to more than one primary source (e.g., a PanSTARRS object may be associated with two *Gaia* objects, but one of those *Gaia* objects may also be associated with a second PanSTARRS object).

  How these different scenarios are treated depends on the parameters below. *One+One* matches are always assigned.

  - `RobustLoResMatch`: If `False` then *Zero/One* matches are allowed. This makes greater allowances for poor astrometry but increases the risk of mismatches.
  - `DeblendLoResMatch`: If `False` then *One/Many* matches are allowed (all potential matches will be assigned). Outlier rejection is then relied on to select the correct match. This works well if one star greatly dominates the blend, but increases both the risk of mismatches and the fit being skewed by blended photometry. No deblending is performed if `DeblendLoResMatch = True`.
  - `DeblendLoResNonMatch`: Not currently used. This is inserted for future compatibility to deal with *Zero/Many* scenarios.
  - `DeblendMinWeight`: Deblending is not currently performed. This is inserted for future compatibility.
  - `DeblendActivationWeight`: Deblending is not currently performed. This is inserted for future compatibility.
  - `MaxBlends`: Currently only used for display purposes, sources with more blends in a catalogue than this value will be highlighted.

- `UseStarsWithNoDist`: If `True`, then sources with no distance, or with distance uncertainties that exceed `MaxDistError`, will be assigned a distance of `DefaultDist`. If `False`, then these sources will be rejected and a warning given.

  The model `adopt_distance` deals with distance calculation. By default, distances are taken from the primary catalogue (`PrimaryCatRef`), but can also come from catalogues specified in the ancillary data input file. Parallaxes are simply inverted to provide a distance, and negative parallaxes are discounted. If more than one distance measure is given, a weighted average is computed, where weights are assigned based on the inverse square of the fractional error. Future versions anticipate a modified selection, similar to the merging criteria defined above. Currently, choices between catalogues can be made using the priority system in the ancillary data input.

- `MaxDistError`: The maximum fractional error in distance allowed before a source is rejected (or given a default distance).

- `DefaultDist`: The default distance, in parsecs, assigned to a source with a measured distance that is missing or rejected.

5.3 Astrometry error treatment in catalogue matching

- `GaiaCone`: For `single`-source and `list` modes of operation, either co-ordinates are entered directly by the user, or co-ordinates are retrieved based on a SIMBAD search. A search for a *Gaia* object near these co-ordinates will be made. This radius (in arcseconds) defines the initial search cone that is used to make that source assignment. If no match is found, a *Hipparcos* match within this radius is attempted. If no match is found, the source will collect what data it can based on a search at the supplied co-ordinates without proper-motion correction (this cannot be later retrieved as part of the ancillary data, though distances can be supplied here). A large `GaiaCone` increases the risk of mismatches, but is more forgiving of astrometry discrepancies between (e.g.) SIMBAD and *Gaia* sources.

- `PosErrMult`: For `single`-source and `list` modes of operation, a multiplier applied to all catalogues' `XMatchCone` values. While control over cross-matching should normally be done on a catalogue-by-catalogue basis in the catalogues input file, this allows a quick adjustment to include or exclude marginal matches. This may be helpful, for example, if the blending or outlier rejection settings are adjusted.

- `AddPMErr`: A fixed addition to the cross-matching radius (`XMatchCone`) that allows for aggregated proper motion errors between two catalogues. Normally this is adviseable, unless the primary catalogue has proper-motion errors that are very large.

- `EpochPMErrYears`: A fixed addition to the cross-matching radius (`XMatchCone`) that allows for proper motion taking place over the survey. The matching radius is increased (*viz.* in all directions!) by the annual proper motion multiplied by this value. Consequently, the most appropriate value to choose is typically the maximum difference in time between the start or end of a survey and the epoch used to represent it (e.g., two years for a catalogue with epoch 2018.0 whose survey lasted between 2016.0 and 2019.0).

5.4 Ancillary data

- `AncillaryFile`: file containing ancillary data sources, see Section 4.

- **Merging data:** when multiple measurements of the same parameter are found and have been assigned the same priority, they will be merged according to the following parameters. The same process is used as in the merging criteria for photometry, although if a parameter is defined by a string then the first entry in the list is taken.

  - `MinAncError`: similar to `MinPhotError`, the fractional error on an ancillary data point will be increased to this value if it is below it, in order that a more representative weight can be assigned.
  - `AncWeightingLimit`: similar to `PhotWeightingLimit`, the minimum weight to trigger a data merge, otherwise the more precise value will be taken.
  - `AncSigmaLimit`: similar to `PhotSigmaLimit`, data will only be merged if they are within this many multiples of the mean value, otherwise the more precise value will be taken.

5.5 Save points and output

The majority of options in this section help configure the outputs defined in Section 6.1, hence only a brief description is given here.

- `UsePreviousRun`: This option restrarts an instance from fixed points partway through its run.

  - The default (`0`) is to begin a new run.

- Option `1`, forces the programme to use the photometry it has downloaded from the previous run, but otherwise re-create the SED(s) (including any matching) and re-run any fitting.
- Option `2` is designed for skipping setup of area search modes, additionally relying on the previously computed cross-matching between catalogue pairs (beginning with `get_areaseds`);
- Option `3` also uses the previous catalogue cross-matching counts (starting from `compile_areaseds`).
- Option `4` in all modes uses the previously created raw SEDs, but still processes the main loop across sources, allowing for extinction correction, fitting and outlier rejection.
- Option `5` is designed for area searches. It ignores the main loop and only produces the initial and final plots.
- Option `6` is identical, but reproduces only the final plots.

- `OutputSeparator:` This denotes the output separator in the master output, indivdual SEDs, individual ancillary data files, and collated H–R and ancillary data files. Any character is valid, although recommended ones are tab (`\t`), comma or pipe (`|`).
- `CmdArgsFile:` This file stores the command-line arguments used in the previous run for recall by the programme when `UsePreviousRun` > `0`.
- `SaveMasterOutput, MasterOutputFile:` Collated output data file toggle and location. See Section 6.1.
- `MaxIndividualSaves:` When an area search is performed and more than this number of objects is found, output `.sed, .anc` and `.png` files for each source are suppressed.
- `SavePhot, PhotFile:` Internal photometry storage file toggle and location for single sources. Must be set if `UsePreviousRun` > `0`.
- `PhotTempDir, AncTempDir:` Temporary directories for internal storage of photometry and ancillary data while the programme is being run. This is retained for reuse, in case `UsePreviousRun` > `0` on the next run.
- `SavePhotXMatch, PhotXMatchDataFile, PhotXMatchCountsFile, AncXMatchDataFile, AncXMatchCountsFile:` Collated cross-matching data file toggle and locations for multiple sources. Must be set if `UsePreviousRun` > `0`.
- `SaveSEDs, SEDsFile, AncFile, AreaSEDsFile, AreaWeightsFile, AreaAncsFile, AreaAncWeightsFile, AreaCompiledSEDsFile, AreaCompiledAncFile, AreaCompiledSourceFile:` Internal data save points. Must be set if `UsePreviousRun` > `0`.
- `SaveEachSED, SaveEachAnc, SEDsDir, AncDir:` Toggles and directories for the final output SEDs and ancillary data files. See Section 6.1.
- `OutputParamFile, OutputAncFile:` Files for the final tables of fitted parameters and ancillary data. See Section 6.1.
- `OutputAppend:` Toggle to denote read/write access to `OutputParamFile` and `OutputAncFile`. When set to `0` (default) a new file will be generated; `1` will append to the end of the existing file; `2` will scan the list of input targets and only process those that are not already in the output files. Option `2` can be especially useful in cases where the code has aborted unexpectedly (e.g., due to a crash or if connection to the internet has been interrupted). A known bug exists: use of options `1` and `2` may cause issues when creating the global plots for area searches. While `OutputParamFile` and `OutputAncFile` should be correctly updated, `MasterOutputFile` will only contain the results of the current run due to potential changes in the number of columns and to ensure the accuracy of its meta-data.

5.6 Plotting options

Many items in this section are discussed alongside description of the output plots. They are discussed in detail in Section 6.2.

- `PlotSEDs, PlotDir, ShowAstrometryInset:` Toggle and directory for output of `.png` graphs of final SEDs and their fits. The `ShowAstrometryInset` flag can be used to toggle the inset graph showing the proper motion and individual source locations. See Section 6.2.
- `MaxSEDPlotFluxRange:` The maximum flux range allowed on the SED. The faint end of the flux axis will be set at the value of the brightest flux divided by this value. This setting be helpful to hide faint points that do not contribute significantly to the SED.
- `MakeAreaPlots, MakePointPlot, PointPlotFile, MakeClosestPlot, ClosestPlotFile, MakeOffsetPlot, OffsetPlotFile, MakePMPlot, PMPlotFile:` Toggles and locations for output plots, as detailed in Section 6.2.

- `PMPlotClipping`: Allows clipping of high-proper-motion sources. For example, if `PMPlotClipping = 99`, then the proper-motion plot will be sized so that up to 1% of sources with the highest proper motions will lie outside the boundaries of the plot. This can be useful in cases where a few high-proper-motion stars obscure observation of clustering in the proper-motion plot.

- `MakeMatchSkyPlot, MatchSkyPlotFile, MakeHRDPlot, HRDPlotFile`: Toggles and locations for the display of matches against sky co-ordinates and the Hertzsprung–Russell diagram, as detailed in Section 6.2.

- `HRDMinTemp, HRDMaxTemp`: The H–R diagram will be clipped to these temperatures, even if stars are found at most disparate temperatures. This can be useful for hiding sources that have unconvincingly high or low temperatures.

- `HRDMinLum, HRDMaxLum`: The H–R diagram will be clipped ot these luminosities, even if fainter or brighter stars exist. This can be useful when including stars with small parallaxes, which may be given unrealistically large distances.

- `MakeXSPlots, LumXSPlotFile, XSSpacePlotFile, XSCorrnPlotFile`: Toggle and locations for various plots of excess flux, as detailed in Section 6.2.

- `LumXSPlotXSRange`: The range of flux excess and deficit to display on the plots. This defines, e.g., the horizontal axis on `LumXSPlotFile` and the colour scaling on `XSSpacePlotFile` and `XSCorrnPlotFile`.

- `LumXSPlotMinPoints`: The minimum number of points for a filter to be included in the three excess plots. If no filter matches these criteria (because the search radius is too small or the fit criteria too stringent), then a warning will be returned and all data will be plotted.

5.7 Fitting parameters and outlier rejection

5.7.1 Outlier rejection

The following list describes the variables used in outlier rejection. A description of the method used follows.

- `MaxOutliers`: The maximum number of outliers to consider.

- `MaxSeqOutliers`: The maximum number of outliers to consider sequentially before aborting.

- `MinOutliers`: The minimum number of points in the SED required to perform outlier detection.

- `MaxFracOutliers`: The maximum fraction of data points to remove as outliers.

- `OutlierTolerance`: The outlier tolerance, as a fraction of the fitted model flux.

- `OutlierChiSqMin`: The minimum factor by which the reduced χ^2 must be reduced for a point to be considered an outlier.

An SED will contain a number of photometric points: provided this number exceeds `MinOutliers`, the SED will be examined for outliers. Outliers can be single points, or a group of points associated with a particular survey, or a series of points coming from a nearby star that may be partially or fully blended in some catalogues. It is therefore important to test several outliers together before declaring the fit good.

Points in the SED are placed in descending order of fractional distance from the model fit (e.g., if the ratio of modelled to observed flux is 0.6, 0.8, 1.0, 1.5 and 2.0, these points will be ordered as 2.0, 0.6, 1.5, 0.8 and 1.0). The first (worst-fitting) point of the SED is removed and the fit re-run. If this improves the reduced χ^2 by a factor greater than `OutlierChiSqMin`, then the point is considered an outlier and removed from the SED. The second worst-fitting point in the SED is then queried, and so on.

However, if removing a point does not improve the reduced χ^2 sufficiently, then both it *and* the following point are considered together. If removing both improves the fit by `OutlierChiSqMin` then both points are removed. Otherwise, the three worst-fiting points remaining in the SED are considered, and so on.

This process continues until one of the following criteria are met:

1. A total of `MaxOutliers` points have been considered.

2. A total of `MaxSeqOutliers` points have been considered sequentially without identifying any outliers.

3. The number of remaining points in the SED drops below `MinOutliers`.

5.7.2 Fitting procedure

Several options exist to adjust which points are used in the fit, and with what weight they are treated.

- `MinLambda, MaxLambda`: Only points between these two wavelengths (in microns) are used in the fit; points outside this wavelength range will be masked.

  At short wavelengths, the balance is between maintaining enough points on the Wien tail of high-temperature sources on the one hand, and mitigating over-correction of interstellar extinction and presence of ultraviolet excess (due to companions or active stellar chromospheres/coronae) on the other hand. At long wavelengths, the balance is between accurate fitting of the Rayleigh–Jeans tail of the spectrum, especially in cool sources, and including infrared excess in the fit. This infrared excess can come either from the star itself or from background light (e.g., zodiacal light or interstellar dust) and is often exacerbated by the large beam size of infrared surveys.

- `UseWeightedTFit, WeightedTSigma, WeightedTPower`: If `UseWeightedTFit = False`, all points are given equal weight in the fit. If `UseWeightedTFit = True`, points are weighted according to the temperature of the fitting model. The blackbody peak of the model will be calculated ($\lambda_{\rm peak}$), and a modified-Gaussian weighting (w_i) will be applied based on the wavelength (λ_i) of each filter, such that

$$w_i = \exp\left(-\frac{(\log\lambda_i - \log\lambda_{\rm peak})^2}{\texttt{WeightedTSigma}^2}\right)^{\texttt{WeightedTPower}} \tag{3}$$

- `ModelCode`: The name of the model code used in the fit. Currently the only model dataset with full support is `bt-settl`. However, with moderately difficulty, it is possible to incorporate other model datasets that are available at the Spanish Virtual Observatory. Details on how to do this are given in the next section.

- `MinDataPoints`: The minimum number of data points that will be passed for model fitting. Objects with fewer data points will be returned without a fit.

- `WeightByErrors, WeightOffset`: When the `WeightByErrors` flag is inactive, the fitting procedure will treat all photometric points as having equal error, and give them equal weight (subject to the other weighting parameters above). This can be useful, as the errors on many photometric measurements do not correspond well to errors in a star's SED (unaccounted-for errors often include the survey's photometric zero-point errors, filter transmission errors, stellar variability and the effects of unseen stellar blending). Alternatively, the errors can be included when `WeightByErrors` is active, which is useful when including photometry with large fractional errors. In this case, an offset can be applied to errors to set an artificial error floor that accounts for these factors. In this case, the individual photometry will be multiplied by a weight

$$w = \frac{\texttt{WeightOffset}^2}{(F/\Delta F)^2 + \texttt{WeightOffset}^2} \tag{4}$$

  where F is the flux and ΔF its error, meaning $F/\Delta F$ is the signal-to-noise.

5.7.3 Model dataset

- `RecomputeModelGrid`: This option can be used to recast the model grid from a series of stellar models into a faster look-up table, for example to add more filters. See details in Section 3.2.2.

- `ModelTeffLo, ModelTeffHi`: Minimum and maximum ranges of models to load (if `RecomputeModelGrid = False`) and recast (if `RecomputeModelGrid = True`). Reducing this range will speed up the programme's run-time, but you will not be able to fit stars outside this range.

- `ModelLoggLo, ModelLoggHi`: Similar values for $\log(g)$.

- `ModelFeHLo, ModelFeHHi, ModelAFeLo, ModelAFeHi`: Similar values for [Fe/H] and [α/Fe]. Since these parameters are not part of the fitting procedure, if models are requested that are outside this range, the programme will fail to produce a fit for that object entirely.

5.7.4 Initial parameters and assumptions

Calculating a correct stellar model atmosphere to fit requires making assumptions about the mass and composition of the star. In general, these have relatively minor effects on the resulting stellar parameters.

- `UseGaiaModelStart`: This option is not currently active.

- `ModelStartTeff, ModelStartLogg, ModelStartFeH, ModelStartAFe`: Default $T_{\rm eff}$, $\log(g)$, [Fe/H] and [α/Fe] at which to start the model fit. Only $T_{\rm eff}$ and $\log(g)$ will be altered by the fitting procedure.

- `PrecomputeLogg:` If `True`, the programme will first fit a blackbody and compute an initial guess for $\log(g)$ based on that blackbody fit. Otherwise, `ModelStartLogg` will be used.
- `IterateLogg:` If `True` then $\log(g)$ will be updated during each fitting iteration. Otherwise, it will only be updated after the final fit.
- `DefaultMass:` The default mass to assume for a star for the $\log(g)$ calculation. `DefaultMass` can be superseded if `UseMSMass` is `True`.
- `UseMSMass:` If `False` then `DefaultMass` is used. If `True` then:
  - If the star is fitted as a main-sequence star ($T_{\rm eff} > 5500$ K or $L < 2\,{\rm L}_\odot$) then it is assumed that $M \propto L^{1/3.5}$ (with the Sun as reference).
  - If the star is an upper AGB star ($T_{\rm eff} > 5500$ K or $L > 2500\,{\rm L}_\odot$) then two masses are calculated: a core mass (from Blöcker 1993) of $M_{\rm core} = (L/62\,200\,{\rm L}_\odot) + 0.487\,{\rm M}_\odot$ and an initial mass (from Casewell 2009) of $M_{\rm init} = (M_{\rm core} - 0.3569)/0.1197\,{\rm M}_\odot$; these two masses are averaged, on the assumption that most upper AGB stars have lost half of their envelope mass.
  - If the star is neither a main-sequence star nor an upper AGB star, a mass of $1\,{\rm M}_\odot$ is assumed.
- `FeHfromZ:` If `False` then `ModelStartFeH` is used. If `True`, then [Fe/H] and [α/Fe] are estimated based on height above the Galactic Plane (Z in kpc), using

$$[{\rm Fe/H}] = \tan^{-1} Z, \tag{5}$$
$$[\alpha/{\rm Fe}] = \frac{\tan^{-1} Z}{5}. \tag{6}$$

- `UseInformedPriors, PriorTMax, PriorTMaxSoft, PriorTMin, PriorTMinSoft:` These options can be used to set informed priors on the temperature of the models being assessed. `UseInformedPriors` is used to toggle this option on and off. No change is applied to the goodness-of-fit statistic between `PriorTMin` and `PriorTMax`. At temperatures below `PriorTMin`, the goodness-of-fit statistic will be modified by a factor

$$\chi^2 = \chi^2_{\rm original} \frac{2}{{\rm Erf}(\sigma) + 1} \tag{7}$$

  where Erf is the error function and

$$\sigma = (\texttt{PriorTMin} - T)/(\texttt{PriorTMinSoft}). \tag{8}$$

  This allows a smooth, tapering bias that increases towards progressively lower temperatures. `PriorTMax` and `PriorTMaxSoft` are used similarly for high temperatures.
- `UsePriorsOnTspec:` When this option is on, the special parameter `TSpec` is used from the ancillary data. `PriorTMax` and `PriorTMin` are set to the merged value of this parameter, and `PriorTMaxSoft` and `PriorTMinSoft` are set to its error. This allows an object with an existing temperature measurement to have its other parameters estimated. This can be a useful process, for example, when fitting extinction towards an object.

5.8 Treatment of interstellar extinction

Extinction correction proceeds via a standard Fitzpatrick (1999) reddening law.

- `DefaultEBV:` A default value for $E(B-V)$ when a value cannot be extracted from the extinction map.
- `DefaultRV:` The $R_{\rm V}$ to be used.
- `ExtMap:` The choice of built-in 3D extinction map to use. Choices are:
  - `None:` The programme will revert to `DefaultEBV` in all cases.
  - `Gontcharov2017:` The Gontcharov (2017) reddening cube on VizieR will be queried. If the object lies outwith this cube, then $E(B-V)$ is assumed to be zero.
  - `GTomo:` The G-Tomo extinction cube will be queried.
- `ExtMaxAngle:` Extinction queries can be computationally expensive (∼0.6 seconds per query). A new extinction query will only be requested if the source RA and Dec differs from the previous object by more than this angular distance (in degrees). Otherwise, the previous value is used.

6 PySSED outputs

6.1 Data files

The output data files represent the primary outputs of the PySSED code, and can be found in the output/ directory. They can be considered in three classes:

- `output.dat` is a global master file, creating a single data output for all objects and all parameters (photometric and ancillary). This is the easiest file to use when dealing with large areas, particularly for complex applications (e.g. machine learning). However, because the data output is often in a sparse array, the file can become quite large.
- `hrd.dat` and `anc.dat` are the global parameter files. These include the data needed to make the H-R diagram, and a list of the ancillary parameters adopted or found for each star.
- the `sed/` and `anc/` directories include output files for each object, listing its photometric points and model fits for the spectral energy distribution, and its aggregated ancillary data, respectively.

Large runs: For runs involving very large areas, it is advisable to turn off reporting of the `sed/` and `anc/` directories, since these include one file per star. What counts as "very large" depends on your datasets, requirements, computing capacity and area of sky, but typically it is best to turn these off if you expect thousands of stars and are therefore looking at an area of $\sim 0.1^\circ$ or more.

Structure: All files are tab-separated, plain-text files. A general rule for the output files is that they start with a set of comments describing their contents and header information. These are commented out with a "#" symbol, so can be read back into Python with `numpy.genfromtxt` using the `comments='#'` option. Control of the output data files is dictated by the setup file, in the section labelled "*Use of save points and output*".

6.1.1 `output.dat`

Purpose: This file aggregates all of PySSED outputs together.

Control: It can be turned on and off, and its location can be changed using the following two lines in the setup file:

```
SaveMasterOutput 1
MasterOutputFile ../output/output.dat
```

Structure: This file begins with two lines of comments containing the PySSED version number and the PySSED command-line parameters that generated it. There are then five lines of header information. These describe:

- The parameter type. These can be:
  - *Object*: the first column, the *Gaia* name of the object (for area searches) or the user-inputted name of the object (for list searches).
  - *Adopted*: adopted "special" parameters based on a combination of *Gaia* and user-inputted data (see below).
  - *Photometry*: fluxes from photometric catalogues.
  - *Dereddened*: the same fluxes, dereddened according to the adopted dereddening model.
  - *Model*: the modelled flux to be compared to the same dereddened data.
  - *Ancillary*: ancillary parameters that do not belong in the "special" category and are simply recorded for aggregative or display purposes.
- The parameter name. These will include:
  - *Object*: as before.
  - "Special" parameters: these are parameters that are used in the fitting process, or are the result of fitting models to the object. These should include:
    - `RA` (right ascension) and `Dec` (declination),
    - `PMRA` and `PMDec` (proper motions),
    - `Teff` (effective temperature),
    - `Lum` (luminosity),
    - `Distance`,
    - `Av` (extinction, A_V),
    - `logg` (surface gravity, $\log(g)$) and
    - `[Fe/H]`.
  - Names of photometric filters in the Spanish Virtual Observatory format.
  - Names given to parameters in the ancillary input file.

- A flag denoting whether that column contains a `Value` or its `Error`.
- The units for that value or error.
- A wavelength column for photometric data, listing the central wavelength ($\lambda_{\rm eff}$) for that filter in Angströms, taken from the Spanish Virtual Observatory. The first column in this line is "*Wavelength[AA]*". Subsequent non-photometric data is marked "-".
- A similar column titled "*Width[AA]*" listing the effective width of the filter in Angströms.
- A similar column titled "*Alambda*" listing the adopted extinction ratio A_λ/A_V.

Each subsequent line denotes the results for an individual object.

Notes: This file will typically contain hundreds of columns. Parsing them can be difficult to do by eye. A useful way of extracting this information in transposed form is with the following command:

```
awk 'NR>=3 && NR<=9 {for (i=1;i<=NF;i++) {n[i]=n[i]" "$i; if (NR==9) print i,n[i]}}' output.dat
```

The `grep` tool can be used to search within this metadata for specific columns of interest, e.g.
`awk ...  output.dat | grep Gaia`
to search for *Gaia* data or
`awk ...  output.dat | awk '`$6 + 0 > 3000.\&\&$`6<8000.'`
to search for any data at optical wavelengths.

6.1.2 `hrd.dat`

Purpose: This file contains the information generated by PySSED and used to create the Hertzsprung–Russell diagram.

Control: This file has no controls. Its location is determined by `OutputParamFile` in the setup file.

Structure: This file has two header lines listing the parameters and their units. The columns respectively list the object ID, computed effective temperature in Kelvin, the luminosity in solar luminosities, the radius in solar radii, and the χ^2 value of the fit.

Notes: Care should be used if using the χ^2 value as a goodness-of-fit determinant. The absolute value of χ^2 will be affected by the choice of how photometric errors and weighting are applied to the photometry in the `compute_model` subroutine. In the standard setup, the photometric errors are ignored and points given equal errors in the fit. This is to reduce the effect of overly precise photometry, which may not account for intrinsic variability or systematic errors in the photometric reduction.

6.1.3 `anc.dat`

Purpose: This file contains any ancillary information gathered by PySSED but not used in the fit.

Control: This file has no controls. Its location is determined by `OutputAncFile` in the setup file. Its contents are defined by the ancillary data input file.

Structure: This file has one header line listing the parameter name as defined in the ancillary data input file. The columns begin with object ID and then list the parameters in their input order. These should typically include right ascension (`RA`), declination (`Dec`), distance, [Fe/H] (`FeH`), parallax and *Gaia* spectroscopic temperature (`Tspec`) and $\log(g)$ (`logg`).

6.1.4 `sed/` directory

Purpose: This directory contains the spectral energy distributions for each object.

Control: Writing to this directory can be turned on and off, and its location can be changed using the following two lines in the setup file:
`SaveEachSED 1`
`SEDsDir ../output/sed/`
These files will not be saved if an area search identifies more than `MaxIndividualSaves` objects in the master catalogue (normally *Gaia*).

Structure: Each object has an associated `.sed` file labelled by its object identifier (*Gaia* source ID for area searches, user input for list or single-object searches). One line of header information lists the column names. Subsequent lines include data for each photometric observation. Columns are:

- `catname`: catalogue name in catalogues input file
- `objid`: object ID in that catalogue;

- `ra`: right ascension compared to fundamental position from primary catalogue (normally *Gaia* or Hipparcos), in arcseconds;
- `dec`: declination, similarly;
- `modelra`: model right ascension compared to fundamental position from primary catalogue, accounting for proper motion, in arcseconds;
- `modeldec`: model declination, similarly;
- `svoname`: Spanish Virtual Observatory (SVO) identifier for the filter;
- `filter`: filter name in the filters input file;
- `wavel, dw`: filter effective wavelength and width from the SVO;
- `mag, magerr`: magnitude and error in the catalogue (computed from flux if necessary);
- `flux, ferr`: flux and error in the catalogue (computed from magnitude if necessary), in Jy;
- `dered, derederr`: dereddened flux and error according to the chosen extinction model, in Jy;
- `model`: flux modelled by the programme when fitting a model (e.g., blackbody or stellar atmosphere model), in Jy;
- `mask`: Boolean flag dictating whether that observation is used in the SED fit.

6.1.5 `anc/` directory

Purpose: This directory contains the ancillary data for each object.

Control: Writing to this directory can be turned on and off, and its location can be changed using the following two lines in the setup file:

```
SaveEachAnc 1
AncDir ../output/anc/
```

These files will not be saved if an area search identifies more than `MaxIndividualSaves` objects in the master catalogue (normally *Gaia*).

Structure: Each object has an associated `.anc` file labelled by its object identifier (as for the `sed/` directory). One line of header information lists the column names. Subsequent lines include data for each ancillary data point. Columns are:

- `parameter`: parameter as named in the ancillary data file;
- `catname`: catalogue as named in the ancillary data file;
- `colname`: column name in original catalogue at CDS;
- `value`: retrieved or computed value;
- `err`: error on `value`;
- `priority`: priority assigned in the ancillary data file for merging or choosing data;
- `mask`: Boolean flag dictating whether data should be used (can be `False` if the data is null, if the data does not meet the criteria in the reject-reasons input file, or if the data has been merged by a `priority` request).

6.2 Image files

6.2.1 `hrd.png`

Purpose: Hertzsprung–Russell diagram of analysed objects

Control: This plot is switched on and off, and its location controlled using the following lines in the setup file:

```
MakeHRDPlot 1
HRDPlotFile ../output/hrd.png
```

The plot is generated in the `globalpostplots` subroutine when an area search is performed. It is not generated for single or list searches.

Contents: This plot displays the well-known Hertzsprung–Russell plot of effective temperature versus luminosity. Points are coloured by distance, from red (closest) to blue (furthest). Point size and opacity varies with the number of stars.

Notes on interpretation: Poor-quality data may result in several artefacts becoming present on the H–R diagram. Examples include:

- The diagram may exhibit vertical striping, particularly away from the main sequence and giant branch. The striping corresponds to the temperatures at which the model atmosphere is defined. It is normally caused in objects when insufficient data exists to obtain a good-quality fit, or sometimes when fitted data is of poor quality. Poor-quality data may include non-stellar objects like galaxies, objects which have been poorly deblended, or may result from mistakes in input (e.g. in the properties of the input filters, or their limits of saturation and accuracy). Examining the SEDs of a few stars within these stripes will help you determine which of these problems is occurring.
- Vertical smearing of the main sequence and giant branch. This is a natural feature of having inaccurate distances. If you know the distance to your stars (e.g., if you are looking at a cluster or galaxy), consider changing the parameters `UseStarsWithNoDist` to 1, `MaxDistError` to a smaller fraction, `DefaultDist` to the distance of your cluster/galaxy; also set `ModelStartFeH`, `ModelStartAFe` and `DefaultEBV` to the [Fe/H] ratio, [α/Fe] ratio and $E(B-V)$ colour excess for your cluster/galaxy, and ensure `FeHfromZ` is zero and `ExtMap` is "*None*".
- Horizontal smearing or splitting (doubling) of the main sequence and giant branch. This can be caused by differences in the wavelength coverage within a dataset, particularly if one or more filters has a zero-point error. This can be diagnosed in the `xs-...png` files.
- Diagonal (top-left to bottom-right) smearing of the main sequence and giant branch. This is normally caused by an inaccurate reddening correction. It will be most pronounced for bluer stars.

6.2.2 `closest.png`

Purpose: This plot shows the angular separation between sources in specific catalogues. It can be useful for setting criteria for cross-matching between surveys, and for diagnosing astrometric problems.

Control: This plot is switched on and off, and its location controlled using the following lines in the setup file:
`MakeClosestPlot 1`
`ClosestPlotFile ../output/closest.png`
The plot is generated by calling the `globalpostplots` subroutine when an area search is performed, and is actioned within the `globalclosestplot` subroutine. It is not generated for single or list searches.

Contents: This plot is separated vertically into individual surveys. For each source in that survey, the distance to the nearest neighbour *in any survey* (including the same survey) is plotted on the horizontal axis. Grey '+' signs denote the median value within that survey and red '×' signs denote the cutoff used for matching. Point size and opacity varies with the density of objects within each survey; point colour varies with the shortest wavelength of observation in that survey. Points are randomly smeared in the vertical direction to form a clearer bar for each survey.

Notes on interpretation: Points to the left of the '×' signs will be accepted as matches; points to the right will be rejected. For most surveys, we expect good correspondence between matches. The '+' signs should be significantly left of the '×' signs, though this may not always be the case for surveys with a small number of observations or at very different wavelengths from the primary survey (usually *Gaia*).

Astrometric offsets can be judged alongside the `offset.png` and `matchsky.png` files. In this plot, they will manifest themselves as a small number of cross-matches at small radii and a large number of cross-matches congregating near a specific larger radius, which may be considerably larger than the physical resolution (beam size) of the survey.

6.2.3 `matchsky.png`

Purpose: This plot shows matching targets between pairs of surveys as a function of position the sky.

Control: This plot is switched on and off, and its location controlled using the following lines in the setup file:
`MakeMatchSkyPlot 1`
`MatchSkyPlotFile ../output/matchsky.png`
The plot is generated by calling the `globalpostplots` subroutine when an area search is performed, and is actioned within the `globalmatchskyplot` subroutine. It is not generated for single or list searches.

Contents: This corner plot shows a sky map for pairs of surveys to be read off the horizontal and vertical axes. For each sky plot, objects from the horizontal survey are shown. They are colour-coded according to their cross-matches on the vertical survey: blue objects have no viable match, green objects have a unique viable match, and red objects have multiple matches.

Notes on interpretation: Most often, features will appear on these plots as sharp blue and red boundaries. Blue regions with sharp boundaries tend to represent regions where the vertical survey has no coverage (entirely blue maps indicate that the vertical survey has few objects compared to the horizontal survey). Red regions with sharp boundaries tend to denote catalogues where objects in some regions are referenced twice (entirely red maps indicate that the vertical survey has poor resolution or too large a matching radius).

Other features in these maps can be caused by a variety of effects. Thin blue lines can be caused by diffraction spikes, while blue spots can be created by bright stars, both of which occur in the vertical survey (holes appear if bright stars exist in the horizontal survey). A gradual fade from green to blue may indicate a problem with the astrometric solution of one or both surveys.

6.2.4 `offset.png`

Purpose: This plot shows the two-dimensional offset between cross-matches.

Control: This plot is switched on and off, and its location controlled using the following lines in the setup file:
`MakeOffsetPlot 1`
`OffsetPlotFile ../output/offset.png`
The plot is generated by calling the `globalpostplots` subroutine when an area search is performed, and is actioned within the `globaloffsetplot` subroutine. It is not generated for single or list searches.

Contents: The plot shows the data from `closest.png`, represented in both RA and Dec; namely, the plot shows the RA and Dec offset to the closest matching object each object has in *any* catalogue (including its own). The points are colour-coded by wavelength, and point size and opacity are controlled by the number of points in that catalogue.

For each catalogue, a median separation for each catalogue is plotted as a grey '+' sign. In cases of good astrometric agreement and large data volumes, these may not be visible among the other points. The extent of the plot is set to three times the distance of the most-disparate catalogue's median, or $0.2''$, whichever is larger.

Notes on interpretation: A plot showing good astrometric agreement will normally have an extent of $0.2''$ and show a strong concentration of stars towards the centre of the plot. If a catalogue has poor astrometry, this will show up as a grouping of a specific colour of points away from the centre, and probably a larger extent. The `closest.png` plot can be used to firmly identify the catalogue (which should be the same colour).

If a published catalogue has poor astrometry, it is advisable to check the equinox and epoch of the photometry. Photometry should be supplied in J2000.0 co-ordinates, with an epoch of the catalogue date, unless the catalogue has been proper-motion corrected to some other date (e.g. 2000).

6.2.5 `pm.png`

Purpose: Show the adopted proper motion for each object.

Control: This plot is switched on and off, and its location controlled using the following lines in the setup file:
`MakePMPlot 1`
`PMPlotFile ../output/pm.png`
The plot is generated by calling the `globalpostplots` subroutine when an area search is performed, and is actioned within the `globalpmplot` subroutine. It is not generated for single or list searches.

An additional parameter, `PMPlotClipping`, controls the percentage of proper-motion vectors shown on the plot (e.g., a percentage of 99 removes the fastest 1% of proper motions). This clipping percentage can be useful to avoid plots dominated by a small number of high-proper-motion stars.

Contents: The plot shows the proper motion in mas yr^{-1} in both the RA and Dec axes. The points are colour-coded by distance as in the H–R diagram.

Notes on interpretation: This plot is useful to visually separate members of clusters and galaxies from foreground or background objects.

6.2.6 `xs-lum.png`

Purpose: This plot identifies excess or deficit in each photometric filter as a function of observed flux. It can be used a means of identifying problems with photometric observations or the assumptions that have been made about the photometric filters that are used.

Control: This plot is switched on and off, and its location controlled using the setup variable:
`MakeXSPlots 1`
which acts on all three excess plots and `LumXSPlotFile ../output/xs-lum.png`
The plot is generated by calling the `globalpostplots` subroutine when an area search is performed, and is actioned within the `excesslumplot` subroutine. It is not generated for single or list searches.

The parameter `LumXSPlotXSRange` sets the horizontal range in this figure, both ranges in `xs-correlation.png`, and the limits of colour-coding in `xs-space.png`. Filters will be ignored in these three plots if they do not have more than `LumXSPlotMinPoints` observed points (unless no filters do, in which case all data will be plotted).

Contents: This file contains a grid of plots, one for each photometric filter. Points that are rejected, either by criteria in the reject-reasons file, limits set in the catalogue and filters file, or by outlier rejection in the fitting procedure, are shown in green. These correspond to the mask == False objects in the SEDs. Points of other colours are colour-coded by temperature, approximating to the anticipated optical colour of their star (e.g., 10 000 K is light grey). The vertical axis is logarithmic and shows the catalogue flux of the star.

Notes on interpretation: In conjunction with the two other excess plots, this plot is particularly useful for identifying problems caused by untrapped errors in photometric data and their photometric filters. Issues that can be diagnosed on this diagram include:

- Flaring in the bottom of the plot which approaches the left- and/or right-hand boundaries. This occurs in catalogues with a low limit to their signal-to-noise ratio for source detection, leading to large uncertainties on faint stars. Sometimes those uncertainties are well-modelled, sometimes they are not. In these cases, the Malmquist bias can become important too. Solutions can include:

  – Removing faint data by imposing a magnitude cut on the offending catalogue in the filters input file. This removes any ambiguity about including the data, but may exclude a large number of stars that are faint but do have good fits.
  – Removing faint data by imposing a colour selection criterion in the reject-reasons file (e.g. if two observations in nearby filters have a colour difference in excess of some limit, a chosen one of them is masked out). This is best practice if suitable filters exist.
  – Leaving these objects in and letting the outlier rejection criteria remove badly fitting objects.

- A left-hand hook at the top of the plot. This indicates poor correction for saturated stars, and is a feature of many surveys. The best solution is normally to invoke a magnitude cut to the bright end of that survey in the filters input file.
- A global offset to the left or right, which may or may not have a colour gradient. This indicates that one or both filters probably has a zero-point error. This may be caused by many issues, including errors in the photometric zero-point, the shape of the filter transmission function, or a need to colour-correct the data.

  Identifying the filter(s) at fault can be difficult, but they will often show up as the strongest anti-correlation (top-left to bottom-right sequence) in the `xs-correlation.png` file.

  This problem can also manifest itself as...

- A split into two vertical sequences, one or both may be offset from the unity line. This occurs when the above problem happens on a catalogue that does not cover the entire observed area. In this case, you should be able to identify the offending catalogue by the shapes it causes in the `xs-space.png` file. Bear in mind that the correlations are all interlinked, so extra information is needed to determine which is the catalogue causing the offset and which is the catalogue being affected.
- A spread of red sources with excess at infrared wavelengths. If the above problems can be ruled out, this may represent real astrophysical phenomenon, including background galaxies or stars with circumstellar dust. However, if the amount of excess correlates with the beam size of the infrared survey, this may indicate that diffuse emission is being included in the photometric measurement, and that the real excess is lower.
- A spread of blue sources with excess or deficit, particularly in the UV (but potentially also in the infrared as a response). This may indicate a poor correction for interstellar extinction, perhaps indicating that a different interstellar extinction law is needed.

Offsets of a few percent (sometimes even tens of percent) reflect the expected uncertainty of photometric zero points. If a star has a large number of data points associated with it, or if these data points occur outside the range of fitted wavelengths (defined by `MinLambda` and `MaxLambda`), then one or two problems on this scale are not likely to catestrophically affect the fit. However, they may affect the global accuracy of the temperature solution by tens of Kelvin.

6.2.7 `xs-space.png`

Purpose: Display flux excess or deficit as a function of position in space.

Control: See `xs-lum.png`. The location of the file is set by `XSSpacePlotFile` and the file is created in the `excessspaceplot` subroutine.

Contents: A grid of plots shows each photometric filter. Grey points show observations that the stellar model fits well; red points show excess flux; blue points show a deficit of flux in that filter. Green points show objects without a model fit.

Notes on interpretation: See `xs-lum.png`.

6.2.8 `xs-correlation.png`

Purpose: Display flux excess or deficit between pairs of catalogues.

Control: See `xs-lum.png`. The location of the file is set by `XSCorrnPlotFile` and the file is created in the `excesscorrnplot` subroutine.

Contents: A corner plot of filter pairs is shown, similarly to `matchsky.png`. Excess and deficit flux is shown in this plot using the same temperature-based colour scheme as `xs-lum.png`, with the exception that only points used in the fit are shown in this plot (`mask == True`).

Notes on interpretation: See `xs-lum.png`.

6.2.9 png/ directory

Purpose: This directory contains image files of the fitted SEDs for easy inspection.

Control: Writing to this directory can be turned on and off, and its location can be changed using the following two lines in the setup file:

```
PlotSEDs 1
PlotDir ../output/png/
```

These files will not be saved if an area search identifies more than `MaxIndividualSaves` objects in the master catalogue (normally *Gaia*).

Contents: Each object has an associated `.png` file labelled by its object identifier (as for the `sed/` directory). Each PNG file shows that object's SED (note both axes are logarithmic).

Each file contains observed SED, shown as coloured points according to wavelength. Grey error bars extend vertically to show the flux error, and horizontally to show the filter width. Points are shown in grey if they are not used in the fit. If computed, the model fit is shown as purple squares, connected by a purple line. This can be used to ensure the correct observations are being used in the fit.

The inset plot shows the same observations plotted on the sky. A purple chord shows the expected track of the star if it follows its adopted proper motion. Purple lines connect the observed astrometric positions back to the position expected at that epoch from the proper-motion model. This can be useful in ensuring the correct star has been selected.

7 Version history

7.1 Version 0.3

7.1.1 Major changes

- 0.3.20230707: Allowed prior constraint on temperature
- 0.3.20230707: Allowed fixed errors in ancillary data
- 0.3.20230620: Allowed different output file formats (TSV, CSV, etc.)
- 0.3.20230523: Allowed files for ancillary queries
- 0.3.20230503: Allowed square and rectangular area searches

7.1.2 Minor changes

- 0.3.20230503: Remove zero-magnitude points
- 0.3.20230503: Allowed line-of-sight extinction to vary over area searches.
- 0.3.20230503: Made insets in SED plots optional.
- 0.3.20230503: Fixed headers in ancillary output file.
- 0.3.20230503: Fixed bug in G-Tomo extinction correction (A_V was reported instead of $E(B-V)$).
- 0.3.20230503: Fixed bug in photometric-error-weighted fitting.
- 0.3.20230503: Fixed bug causing NaNs when error is zero.

7.1.3 Known features

- 0.3.20230314: Bug related to astroquery v.0.4.6 or its dependencies where RA/Dec co-ordinates are sometimes improperly parsed. Cannot reproduce on test systems.
- 0.3.20230314: Error occurs if Python does not recognise NoneType and encounters a NoneType error.

References

Fitzpatrick E. L., 1999, http://dx.doi.org/10.1086/316293 , https://ui.adsabs.harvard.edu/abs/1999PASP..111...63F 111, 63