Title: LOR: The Phosphoros template-fitting code
Contributors: Will Hartley, Stephane Paltani
Co-signers: Florian Dubath, Alejandro Alvarez Ayllon, Guillaume Desprez

0. Summary Statement

Phosphoros (https://anaconda.org/astrorama/phosphoros) is a new, flexible and fully Bayesian template-fitting photo-z code developed for use within the Euclid photometric redshift pipeline and to fulfill the need to have professionally-written code that is supported throughout the mission's lifetime. It is written and maintained by the Euclid Swiss Science Data Centre.

The code took as a starting point the feature set of the widely-used Le Phare package, but now includes several novel features that takes its utility beyond its spiritual predecessor. Some of the features listed here are detailed further in the relevant sections below.

- The code can be run either from the command line or via its graphical user interface. The concept for the GUI is that a user can test and adjust their set-up easily and intuitively and then use the config file that it creates for production runs from the command line or via scripts.
- One aspect of the GUI that is designed to assist the user in optimising their set-up is an automatic post-processing of the results, if a spectroscopic redshift column is defined in the input catalogue. The post-processing produces statistics, performance metrics and comparison plots of photo-z vs spec-z, and allows the user to inspect the resulting PDFs.
- Contrary to the most common template-fitting codes, Phosphoros is fully Bayesian. Arbitrary priors can be applied on any dimension of the grid, including the spectral energy distribution (SED) axis. Joint priors (on more than one dimension) can be constructed, and a joint prior on luminosity (combining SED, redshift and normalization) is proposed by default. Phosphoros can output multidimensional posteriors, or marginalize over any dimensions, as well as provide maximum-likelihood estimates, if the user wants.
- Phosphoros can compute a set of weights for the input SEDs to compensate for the implicit prior that would otherwise be caused by over or under-sampling a particular region of colour space with SEDs. This creates an (approximately) uninformative prior on the SED.
- Accounting for the impact of Milky Way redding is performed following Galametz et al. (2017), whereby the SED is reddened rather than the noisy data de-reddened (which is ill defined for low S/N or negative fluxes). Phosphoros therefore takes the best estimate of the intrinsic SED of the object into account to compute the Galactic reddening.
- Variation of filter response curves across the focal plane can be taken into account in Phosphoros by way of defining the effective shift of the filter mean wavelength (see Paltani et al., in prep.). This is shown to remove a contribution to the bias in the photometric redshifts that is of the order of the total acceptable bias for the Euclid and LSST cosmology probes.
- Physical properties associated with SEDs (e.g. galaxy type, mass-to-light ratio, age, metallicity) can be declared in the header information of a set of SEDs. These properties are then available to Phosphoros and can be output for each object. In the cases where the property naturally scales with the galaxy luminosity (e.g. M/L to produce a stellar mass), they can be declared as such and will be output as that scaled quantity. As a result, Phosphoros can seamlessly provide one-dimensional or multidimensional probability distribution functions of any physical parameters that are declared in the SEDs.

Phosphoros also includes a number of features in common with Le Phare or other template-fitting codes, generally with improvements over the initial implementation:

- Template-fitting at fixed spectroscopic or reference redshift.
- Addition of emission lines to an SED, based on its rest-frame UV. In Phosphoros, this can be freely defined by the user.
- Interpolation between templates.
- Internal dust reddening, with a common or user-defined dust law.
- Luminosity prior, via Schechter function or any tabulated luminosity function. In addition, a volume prior can also be set if the user does not want to use a luminosity function.
- Automatic adjustment of photometric zero-points to improve performance.
- On-the-fly transformation of photometric uncertainties to deal with underestimated uncertainties, to add systematic uncertainties or to add shot noise.
- IGM absorption, with several implementations taken from the literature.

The code is already available as a beta release and has already been used in some publications, although it is currently in the testing stage of development. A formal release of v1 with a presentation paper is expected before the end of this year (2021).

1. Scientific Utility

Phosphoros has wide utility, and is not restricted to any particular science case(s). The ability to output physical properties of galaxies makes it particularly suited to statistical studies within galaxy evolution, or wherever such properties enhance the scientific output - e.g., galaxy type or mass-dependent LSS measurements, cosmology with type 1a supernovae. The template interpolation functionality allows the user to combine galaxy SEDs with AGN SEDs and therefore infer the fraction of light in each component. The interpolation can be restricted to any number of pairs of SEDs at the user’s discretion.

The fact that Phosphoros can be run at a fixed reference redshift means that it is also able to be used for stellar type classification, or could be used to provide value added outputs such as physical properties at some reference redshift - e.g. a spectroscopic redshift. Phosphoros is not restricted in the number or wavelength of photometric bands that are used, provided suitable template SEDs are available. It can handle both photon-counting and energy-counting bandpasses simultaneously. Science cases that have extremely high accuracy requirements on the redshift bias of PDF or ensemble distributions may need additional calibration of the Phosphoros outputs.

2. Outputs

Internally, Phosphoros produces multidimensional likelihoods and posteriors. The outputs of Phosphoros are flexibly configurable, and the likelihoods and posteriors can be projected or marginalized, respectively, on any (set of) dimension(s), in order to produce one-dimensional posteriors and/or likelihood distributions on a user-defined redshift grid, as well as distributions for SED type and dust reddening. The maximum likelihood and best posterior model can also be output. Flag information is output in the cases of missing data, or if upper limits are used (the use of upper limits is configurable).

In addition, Phosphoros can be configured to return a number of samples (chosen by the user) from the full multi-dimensional posterior distribution. Phosphoros achieves this by first computing the likelihood on the grid of SEDs, redshifts and internal reddening (if required), and then importance sampling the full space via interpolation between these grid points. As the SEDs can be tagged with physical properties, those properties will also be output, providing the user with a set of samples from the redshift-stellar mass-age-metallicity distribution, or for whichever set of properties they choose to define. Posterior samples are a compact way of representing the multi-dimensional posterior distribution, making it feasible to store the information while also preserving the correlations between properties that are necessary for accurate scientific analyses.

3. Performance

An early version of the Phosphoros code was used as part of a Euclid photo-z data challenge (Euclid consortium, Desprez et al. 2020). In terms of the key Euclid photo-z performance metrics, its results were on par with Le Phare (the best performing of the 13 codes tested), even though many of the advanced features of Phosphoros were not used. In the point estimate metrics of σ_z and outlier fraction, Phosphoros fell slightly short of the LSST road map requirements (https://dmtn-049.lsst.io/), averaged over all redshifts. However, it should be noted that the Euclid data challenge was run on shallower data than LSST will have and without u-band photometry.

Any template-based photo-z code will be limited in performance by the SED library used, and obtaining an SED set that can meet the LSST requirements across the full redshift range is an unsolved problem at present. We recommend that Phosphoros is tested on commissioning data during the Photo-z Validation Cooperative and that efforts are put towards developing the necessary SED set (this is clearly a cross-community task).

4. Technical Aspects

Scalability - Will probably not meet. Template-fitting codes are typically computationally expensive, relative to many machine learning methods, unless the number of SEDs and redshift sampling are heavily restricted. Presently, the role foreseen for Phosphoros within the Euclid photo-z pipeline is to produce the posterior distributions for a set of reference galaxies, which are then used in conjunction with a nearest-neighbour algorithm to produce the stored output for the main sample. Another alternative could be to forego storage of posteriors for the main LSST sample entirely and generate them on-the-fly as the user requests them (either server side or client side). This is the concept behind the GALPRO software (Mucesh et al. 2021).

Inputs and Outputs - Will meet. Phosphoros requires fluxes, flux uncertainties and optionally the value of Milky Way reddening at the location of the target objects. Output PDFs and point estimates are consistent with the object elements. Additional quantities (e.g. percentiles) can be obtained via post-processing of the PDFs, and is already implemented within Phosphoros.

Storage Constraints - Will meet. Phosphoros requires a library of template SEDs, in ascii format, similar to any template fitting code. SEDs and auxiliary products are expected to take up a few hundred MB.

External Data Sets - Will meet. A sample of objects with spectroscopic redshifts is extremely useful for development, but not required at runtime.

Estimator Training and Iterative Development - Will probably meet. A lot of effort is required in the coming year to construct a suitable SED set for template-fitting photo-z and to develop methods to remove residual redshift biases. However, there is no training phase required.

Computational Processing Constraints - Will meet. Phosphoros is trivially parallelizable, allowing the memory use to be kept small.

Implementation Language - Will meet. Phosphoros is written in C++, with some optional parts (e.g., PDF post-processing) in Python.