SPARCL (Beta) Users Guide

SPARCLUserManual.pdf
Last updated: 23 November 2022
Server API Version: 8.0
Client Version: 1.0.0

Table of Contents

What is SPARCL?

SPARCL (SPectral Analysis and Retrievable Catalog Lab) is an online service for discovery and retrieval of one-dimensional optical-infrared spectra. SPARCL is designed to work for large survey datasets containing many millions of spectra, and to provide access to multiple different data sets through common methods.

SPARCL consists of three main components:

- A database of spectra and metadata
- A server that provides web-service access to the database
- A client package for Python-based data discovery and retrieval

SPARCL currently contains one-dimensional spectroscopic data from the Sloan Digital Sky Survey (SDSS), from both the original SDSS spectrograph and the upgraded instrument of the Baryon Oscillation Spectroscopic Survey (BOSS). SPARCL has been designed and tested to support spectra from the Dark Energy Spectroscopic Instrument (DESI), which will be included in the database after they have been released publicly.

The SPARCL client can be installed in a user's local computing environment, and SPARCL can also be used from within NOIRLab's Astro Data Lab. The `sparcl.main` table within the Astro Data Lab database enables users to connect complex catalog-driven science queries to science-quality spectra.

SPARCL is currently in "beta" release. Users are encouraged to submit questions, bug reports, and feedback to datalab-spectro@noirlab.edu

Other documentation and training resources

In addition to this user guide, the following resources also provide information on SPARCL and the data within it:

- SPARCL website home page
- SPARCL python client documentation at readthedocs
- SPARCL client release history at pypi.org
- SPARCL introductory tutorial Jupyter notebook
- Sloan Digital Sky Survey
- Dark Energy Spectroscopic Instrument (DESI)

Installing and starting the SPARCL client

The client package is pre-installed in the NOIRLab Astro Data Lab. You can also install the latest release of the SPARCL python client software on your own system via

```
pip install sparclclient
```

The client can be loaded within a Python session or program via

```
>>> import sparcl.client
>>> client = sparcl.client.SparclClient()
```

Data sets currently available through SPARCL

The following data sets are currently available through SPARCL:

Data Set Name	Description
SDSS-DR16	One-dimensional optical spectra from the original 640-fiber SDSS spectrograph, as of SDSS Data Release 16
BOSS-DR16	One-dimensional optical spectra from the upgraded 1000-fiber BOSS spectrograph, as of SDSS Data Release 16

Available data sets are also listed on the SPARCL home page. Data from the Dark Energy Spectroscopic Instrument (DESI, funded by the Department of Energy Office of Science) will be available within SPARCL once they are released publicly. This manual will be updated as new datasets are added.

Types of data available through SPARCL

Each record within the SPARCL database corresponds to a single spectrum and includes various different fields. SPARCL categorizes these fields into three different kinds:

CORE fields	Basic parameters that are (ideally) defined for all spectra in the database, and that can be used for data discovery. These are parameters such as RA, Dec, exposure time, instrument, etc.
SPECTRA fields	Vector fields for spectral flux data and related data such as wavelength and inverse variance. These are standardized across data sets as much as possible, however some data sets may include unique SPECTRA fields that do not appear in all data sets.
AUX fields	Additional parameters associated with a spectrum that are not standardized across data sets and cannot be used for data discovery.

The current catalog of available CORE, SPECTRA, and AUX fields, including their units, can be found in the online SPARCL Field Catalog page. Appendix A below also lists the current CORE fields. The provenance of all SPARCL fields from the original data providers is documented in Appendix B below.

Using the SPARCL client for data discovery

CORE fields can be used for data discovery through the `client.find()` method.

A call to `client.find()` can include constraints on any of the CORE fields via the `constraints` argument, which accepts a dictionary with field names as the keys and the constraint conditions as the values. Depending on the field, the constraint conditions can either be (a) a list of desired values or (b) a range of desired values. The type of constraint applicable to each CORE field is shown in the table in Appendix A below. See the SPARCL demo notebook for example usage.

The `client.find()` method returns a results-object which has an attribute called `ids` that is a list of the unique identifiers of all records in the SPARCL database that satisfy the supplied data-discovery conditions. This list of identifiers can then be used to retrieve spectra as described below in the section on “Using the SPARCL client for data retrieval.”

To aid in refining data-discovery queries, the `client.find()` method can also return any desired CORE field values for the records that satisfy the constraint conditions. A list of names of fields to return can be provided to `client.find()` through the `outfields` keyword argument. The values of these fields can be accessed through the `records` attribute of the returned object, e.g. as follows:

```
>>> found = client.find(constraints=dictionary_of_constraints,
>>>                      outfields=list_of_core_fields)
>>> redshift_of_zeroth_spectrum = found.records[0].redshift
```

Using Astro Data Lab for data discovery

SPARCL’s `client.find()` method provides basic stand-alone data discovery. Users may wish to select spectra using more complex queries or other joined catalog tables. To support these use cases, a table of CORE values for all records in the SPARCL database is mirrored within NOIRLab’s Astro Data Lab science platform as the `sparcl.main` table. See the SPARCL demo notebook for an example of SPARCL data discovery and access via Astro Data Lab.

Using the SPARCL client for data retrieval

Given a list of unique spectrum identifiers found from either the `client.find()` method in SPARCL or the `sparcl.main` table in Astro Data Lab, the `client.retrieve()` method provides access to the spectra themselves.

You can specify the fields to be included in the results of `client.retrieve()` via the `include` keyword argument. Including fewer fields generally gives faster performance. The SPARCL Field Catalog page indicates which fields are returned by default (only a minimal set), and also which fields are returned by specifying `include=’ALL’` (which includes all scientific fields, but excludes some internal SPARCL bookkeeping fields.)

You can verify the availability of fields for a particular data set using the `client.get_available_fields()` method, which takes a list of `data_release` values as its argument. If more than one value of `data_release` is provided, the method will return a list of fields common to them all.

The output of `client.retrieve()` is a results-object, which includes the list of records returned in the `records` attribute. See the example notebook for more details.

```
>>> results = client.retrieve(found.ids, include=list_of_fields,
>>>                           dataset_list=list_of_data_releases)
>>> flux_of_zeroth_spectrum = results.records[0].flux
```

Note about ordering of results: For performance reasons, the list of retrieved records is not necessarily sorted to match the list of input ids. If needed, users should make their own cross-match on ids between inputs and outputs to `client.retrieve()`. The SPARCL demo notebook includes a simple example of this.

Note about specifying datasets: the `dataset_list` argument is not necessary to identify records uniquely in the SPARCL database, since the SPARCL id of a spectrum is globally unique across all data sets. However, the `dataset_list` argument is necessary if your requested list of fields to `include` in the results contains fields that are not available across all data sets.

Note about missing/invalid ids: if the list of ids passed to client.find() includes invalid ids, the `client.retrieve()` will return with a warning and no data records. The method `client.missing()` can be used to identify which ids in the argument list are missing from the SPARCL database. An example can be found in the SPARCL demo notebook.

Retrieving spectra by dataset-specific “specid”

Spectra can also be retrieved using “specid” identifiers as provided by the original surveys and projects that produced the data. The syntax is the same as for `client.retrieve()`, with the list of SPARCL ids replaced by a list of integer “specid” values:

```
>>> results = client.retrieve_by_specid(list_of_specids,
>>>                                     include=list_of_fields,
>>>                                     dataset_list=list_of_drs)
```

Currently, SPARCL only contains SDSS-DR16 and BOSS-DR16. For these data sets, “specid” is equal to the “specobjid” parameter provided by the SDSS project, which is globally unique across all SDSS and BOSS spectra. This provides a convenient mechanism for “direct access” to SDSS and BOSS spectra within SPARCL that does not require either an Astro Data Lab query or a call to SPARCL’s `client.find()`.

Note however that in the future as additional data sets are included within SPARCL, “specid” values will not necessarily be unique across data sets or even within data sets, and `client.retrieve_by_specid()` will return all records having any of the requested “specid” values across all data releases in the `dataset_list` argument (or across the entire SPARCL database if there is no `dataset_list` provided.)

Credits and feedback

SPARCL has been funded by the US National Science Foundation (NSF), and is a service of the Community Science and Data Center (CSDC) at NSF's NOIRLab.

If you use SPARCL in your published research, please include the following acknowledgement text:

> *This research uses services or data provided by the Astro Data Lab at NSF's National Optical-Infrared Astronomy Research Laboratory. NOIRLab is operated by the Association of Universities for Research in Astronomy (AURA), Inc. under a cooperative agreement with the National Science Foundation.*

Usage of the data hosted within SPARCL should be acknowledged according to the guidelines of the respective data providers:

- How to cite SDSS / BOSS data

Questions, bug reports, and feedback on SPARCL can be sent to: datalab-spectro@noirlab.edu

Appendix A: SPARCL Core Fields

Field name	Description	Constraint type
`id`	Universally unique identifier for spectrum record in SPARCL database (string)	List of values (but not intended for data discovery)
`specid`	Spectrum identifier from dataset provider (integer, dataset-dependent)	List of values
`targetid`	Photometric target identifier from dataset provider (integer, dataset-dependent)	List of values
`data_release`	Data set name (string)	List of allowed values from SPARCL Categoricals
`datasetgroup`	Name for group of data sets that go together	List of allowed values from SPARCL Categoricals
`ra`	Right ascension of spectrum in decimal degrees (floating point)	Range of values (may not "wrap" around RA=0)
`dec`	Declination of spectrum in decimal degrees (floating point)	Range of values

`redshift`	Redshift as measured / reported by dataset provider (floating point)	Range of values
`redshift_err`	Redshift error as measured / reported by dataset provider (floating point)	Range of values
`redshift_warning`	Redshift / classification warning flag as reported by dataset provider (integer, dataset-dependent, nominally zero is “good” and non-zero is “bad”)	List of values
`spectype`	Spectroscopic classification as measured / reported by dataset provider (string)	List of allowed values from SPARCL Categoricals
`instrument`	Instrument used to acquire spectrum (string)	List of allowed values from SPARCL Categoricals
`telescope`	Telescope used to acquire spectrum (string)	List of allowed values from SPARCL Categoricals
`site`	Observatory site at which spectrum was taken	List of allowed values from SPARCL Categoricals
`specprimary`	Flag indicating whether or not a spectrum is the “primary” or “best” observation of an object within a specific `data_release` or `datasetgroup` (integer / boolean, 1 for “primary” and 0 for not primary, dataset-dependent)	List of values (but typically would only include 1 if being used for data discovery constraints)
`wavemin`	Minimum wavelength covered by spectrum (Angstroms, floating point)	Range of values
`wavemax`	Maximum wavelength covered by spectrum (Angstroms, floating point)	Range of values
`dateobs_center`	Midpoint time of observation as reported by dataset provider (date-time string)	Range of values
`exptime`	Exposure time of spectrum (seconds, floating point)	Range of values
`updated`	Date of most recent ingest or update of this record in the SPARCL database (date-time string)	Range of values

Appendix B: Provenance of fields in SPARCL

The provenance of each of the fields from the original data providers for each data set are shown below.

BOSS-DR16 and SDSS-DR16: CORE fields

Field name	Comments	File, HDU	Provenance field name
data_release	Is categorical	None	None
datasetgroup	Is categorical	None	None
dateobs		spPlate, HDU 0	DATE-OBS
dateobs_center		spPlate, HDU 0	DATE-OBS
dec		spZbest, HDU 1	PLUG_DEC
exptime		spPlate, HDU 0	EXPTIME
id	Generated	None	None
instrument	Is categorical	None	None
ra		spZbest, HDU 1	PLUG_RA
redshift		spZbest, HDU 1	Z
redshift_err		spZbest, HDU 1	Z_ERR
redshift_warning		spZbest, HDU 1	ZWARNING
site	Is categorical	None	None
specid		specObj, HDU 0	SPECOBJID
specprimary		specObj, HDU 0	SPECPRIMARY
spectype	Is categorical	spZbest, HDU 1	CLASS
targetid		specObj, HDU 0	BESTOBJID
telescope	Is categorical	None	None
wavemax		spZbest, HDU 1	WAVEMAX
wavemin		spZbest, HDU 1	WAVEMIN

DESI-EDR: CORE fields

Field name	Comments	File, HDU	Provenance field name
data_release	Is categorical	None	None
datasetgroup	Is categorical	None	None
dateobs	Is None	None	None
dateobs_center	Is None	None	None
dec		camcoadd, HDU 1	MEAN_FIBER_DEC
exptime		camcoadd, HDU 1	COADD_EXPTIME
id	Generated	None	None
instrument	Is categorical	None	None
ra		camcoadd, HDU 1	MEAN_FIBER_RA
redshift		zall-pix-fuji, HDU 1	Z
redshift_err		zall-pix-fuji, HDU 1	ZERR
redshift_warning		zall-pix-fuji, HDU 1	ZWARN
site	Is categorical	None	None
specid		camcoadd, HDU 1	TARGETID
specprimary		zall-pix-fuji, HDU 1	ZCAT_PRIMARY
spectype	Is categorical	zall-pix-fuji, HDU 1	SPECTYPE
targetid		camcoadd, HDU 1	TARGETID
telescope	Is categorical	None	None
wavemax	Manually added	None	None
wavemin	Manually added	None	None

BOSS-DR16 and SDSS-DR16: SPECTRA fields

Field name	Comments	File, HDU	Provenance field

			name
flux		spPlate, HDU 0	
ivar		spPlate, HDU 1	
mask		spPlate, HDU 2	
model		spZbest, HDU 2	
sky		spPlate, HDU 6	
wavelength	10**(CRVAL1 + (1000 * COEFF1))	spPlate, HDU 0	CRVAL1, COEFF1
wave_sigma		spPlate, HDU 4	

DESI-EDR: SPECTRA fields

Field name	Comments	File, HDU	Provenance field name
flux		camcoadd, HDU 4	
ivar		camcoadd, HDU 5	
mask		camcoadd, HDU 6	
model		camcoadd, HDU 8	
sky	Is None	camcoadd, HDU 9	
wavelength		camcoadd, HDU 3	
wave_sigma		camcoadd, HDU 10	

BOSS-DR16 and SDSS-DR16: AUX fields

Field name	Comments	File, HDU	Provenance field name
FIBERID		spZbest, HDU 1	FIBERID
MJD		spZbest, HDU 1	MJD

PLATE		spZbest, HDU 1	PLATE
RUN1D		specObj, HDU 0	RUN1D
RUN2D		specObj, HDU 0	RUN2D
SPECOBJID		specObj, HDU 0	SPECOBJID

DESI-EDR: AUX fields

Field name	Comments	File, HDU	Provenance field name
FIBER	[DEPRECATED]	camcoadd, HDU 2	FIBER
LOCATION	[DEPRECATED]	camcoadd, HDU 2	LOCATION
HEALPIX		zall-pix-fuji, HDU 1	HEALPIX
MJD	[DEPRECATED]	camcoadd, HDU 2	MJD
PROGRAM		zall-pix-fuji, HDU 1	PROGRAM
SURVEY		zall-pix-fuji, HDU 1	SURVEY
SV_PRIMARY		zall-pix-fuji, HDU 1	SV_PRIMARY