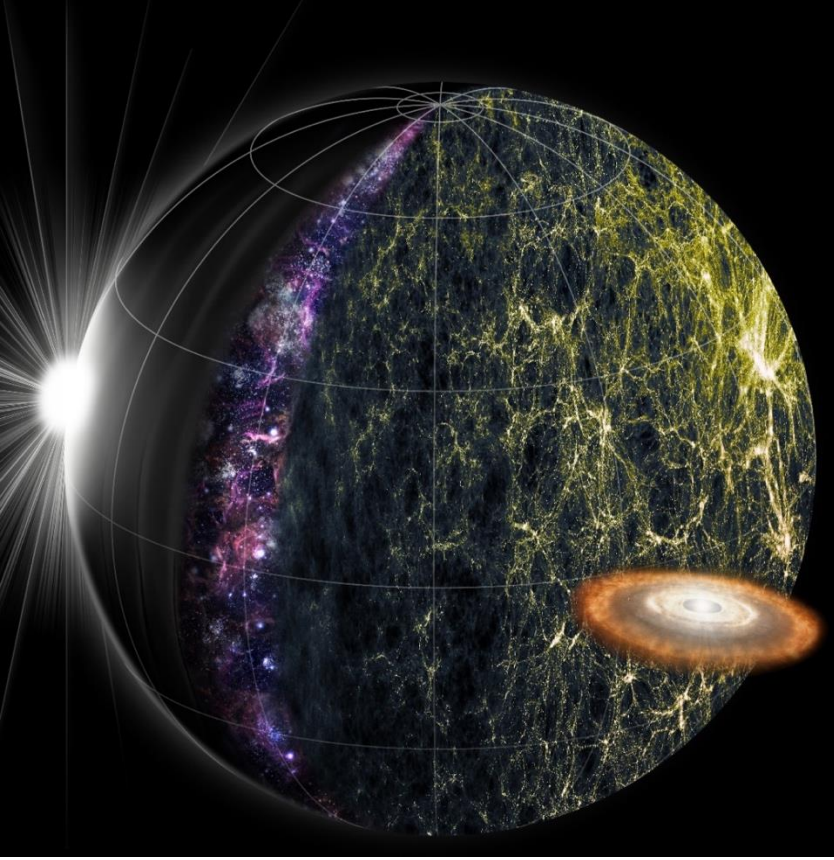


SPHEREx

Jet Propulsion Laboratory
California Institute of Technology

SPHEREx@IRSA: Handling bulk spectra with IVOA interfaces

Gregory Dubois-Felsmann
SPHEREx pipeline system designer

June 11, 2026

What is SPHEREx?

- NASA Medium-Scale Explorer (MidEx)
  - Launched March 2025, science operations began May 1, 2025
  - Managed out of Caltech (Jamie Bock, PI) and JPL (Beth Fabinsky, project manager)

- All-sky spectral survey from 0.75 to 5.0 um, R varying from ~34-128, in 6.15” pixels
  - Six H2RG 2K*2K detectors with linear-variable filters (LVFs) covering the bandpass; passively cooled
  - 102-channel spectra are obtained by observing each location on the sky at 17 locations on each filter
  - Two-year primary mission; sun-synchronous polar orbit covers the sky completely every 6 months
  - ~250 sq. deg. of deep-field coverage at about 50x the all-sky survey, at the NEP and near the SEP
  - Ad-hoc TOO capability enabled observing 3I/ATLAS twice in 2025

- Data processing at Caltech/IPAC, public data release through NASA/IRSA
  - Level 1-3 pipeline software written and operated by Caltech/IPAC (Rachel Akeson, task lead)

What are the SPHEREx Level 2/3 data products (current and future)?

- Level 2 calibrated images
  - The actual images collected through the LVFs
  - Spectral bandpass varies across each image

- Source spectra
  - Obtained via *forced PSF photometry* on LVFs based on a project-internal ”Reference Catalog”
  - Integrates data from Gaia, DESI-LS, Pan-STARRS1, WISE, 2MASS, CALSPEC calibration sources
  - “Native” spectral data at their as-observed central wavelengths
    - NB: These are collected over extended time intervals: 1-3 weeks every 6 months; N_{meas} varies per source
  - Spectra binned to the 102 SPHEREx reference spectral channels

- All-sky cube in the 102 reference channels

We will also be releasing multiple Level 4 catalog and intensity-map data products, ~1 year after the completion of primary mission data collection, produced by the SPHEREx Science Team.
These are not covered in this talk.

What is / will be available in the public data interface at IRSA?

- "Quick Release" data – within 60 days of data collection (available now)
  - Preliminary versions of the Level 2 calibrated images
- Annual cumulative Data Releases (first one to be released November 2026)
  - Level 2 LVF images
  - All-sky cubes in the 102 SPHEREx spectral channels (December)
- High-Reliability Source Catalog (first one to be released mid-2027, again with DR2)
  - Level 3 forced-photometry spectra ("native" and binned)
  - Limited to sources passing quality cuts and stable over at least three seasons of observations
  - Scale is TBD but likely to be at least ~few x 100-million sources, growing in DR2
- Calculation-on-demand science tools (available now – more will come later)
  - Spectrophotometry at user-specified locations
  - Creation of local mosaicked data cubes
  - Downloadable Jupyter notebooks for other data manipulations

Data are also available on NASA's Fornax platform, with open cloud access to the data files.

Things of interest in a community-standards and software-reuse context

NOT IN THIS TALK – but feel free to ask us about these items

- Observation data modeling
  - ObsCore and CAOM2 image metadata, available via public SIAv2 and ObsTAP
  - Aspects of the Level 2 file formats are shared with the Vera C. Rubin Observatory (esp. image masks)
  - Aggressive use of somewhat exotic corners of standard FITS (2D spectral WCS in LVF images)
- Calculational service deployment
  - Services at IRSA are internally deployed as UWS front-ends to command-line tools built from the pipeline software; worker processes run in Kubernetes
- Pipeline construction
  - Rubin Butler and PipelineTask/QuantumGraph/BPS middleware used throughout

Today's standards topics

Theme: handling bulk spectral data

- Main topic: representation of and access to an O(1B)-object catalog of spectra
- Appendix: all-sky spectral cube data format (added after the HiPS 2.0 presentation)

Designing access to the SPHEREx spectral catalog data

- How to provide access to O(1B) spectra?
- Need to support:
  - Access to both the binned / time-averaged spectra and the native variable-length data
    - At the end of the primary mission, every target on the sky should have at least 408 individual measurements in the native form
    - Deep-field sources will have 20,000 individual measurements or more in the “native” spectra
  - Queries for individual spectra
  - Queries for large numbers of spectra
  - Bulk download for broad sky regions, or the entire sky
  - Association of SPHEREx spectra
    - With their actual external progenitors for the forced-photometry coordinates – e.g., Gaia
    - Via general cross-match
  - Ease of use from Python

- We want to provide rich metadata that make the spectra as self-describing as we can

- IRSA (and the SPHEREx project) are committed to using standard formats and protocols where available and to pursuing standards where they are not

The SPHEREx spectral catalog data model

- We will natively compute spectra one HiPS-order-6 tile at a time (about 0.85 sq. deg.)
  - 49,152 tiles
  - Source density varies a lot over the sky (astrophysics + variations in the upstream catalogs' coverages)

- The native file format is Parquet, one row per source, with a mix of:
  - Source-level attributes in scalar columns (IDs, reference coordinates, shape parameters used, etc.)
  - Spectra in vector-valued columns (wavelength, flux, time, numerous additional attributes)
  - Binned spectra have a single common x-axis but are otherwise very similar
  - Tentatively, native and binned spectra will be in separate files
    - For sources covered by the deep fields, we are likely to provide a binned spectrum from the full data as well as a binned spectrum based only on the "underlying" all-sky survey data

- We'll use VOParquet and we want to use a standardized data model for bulk spectra

Annotation of the Parquet spectral data model

- SPHEREx intends to use the VOParquet VOTable-in-Parquet serialization together with an evolution of SpectrumDM to annotate the spectral catalogs “at birth” in the pipelines
  - Just as we create our own ObsCore and CAOM metadata that we hand over to IRSA with the data

- SpectrumDM 1.2 is not directly applicable to our use case of tables of sources, with spectra contained in individual rows

- We’ve previously described a closely related data model which handles multiple spectra:
  - utype=“ipac:MultiSpectrum”

MultiSpectrum

- Already in use as the internal data format for the IRSA SPHEREx Spectrophotometry Tool
  - Currently materialized as XML VOTable but equally representable in VOParquet
  - Table of targets, one row per target
  - Mix of scalar and vector (spectral) data per target
  - SpectrumDM 1.2 Utypes used for annotations, with some tweaks

MultiSpectrum: Header

```
<?xml version='1.0' encoding='utf-8'?>
<VOTABLE xmlns="http://www.ivoa.net/xml/VOTable/v1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.4"
xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.3 http://www.ivoa.net/xml/VOTable/VOTable-1.4.xsd">
 <RESOURCE type="results" utype="ipac:MultiSpectrum">
  <INFO ID="source_coverage" name="SourceCoverage" value="All user provided sources have SPHEREx coverage." />
  <TABLE utype="ipac:MultiSpectrum">
```

MultiSpectrum: Scalar columns (one value per target)

```
<FIELD ID="source_id" datatype="long" name="source_id" ucd="meta.id;meta.main">
   <DESCRIPTION>Unique id of the target in this job</DESCRIPTION></FIELD>
<FIELD ID="label" arraysize="8" datatype="char" name="label" ucd="meta.id">
   <DESCRIPTION>User-specified label of the target ('Target_NNNN' if no name provided)</DESCRIPTION></FIELD>
<FIELD ID="ra" datatype="double" name="ra" precision="7" ucd="pos.eq.ra;meta.main" unit="deg" width="11">
   <DESCRIPTION>R.A. of the target</DESCRIPTION></FIELD>
<FIELD ID="dec" datatype="double" name="dec" precision="7" ucd="pos.eq.dec;meta.main" unit="deg" width="11">
   <DESCRIPTION>DEC of the target</DESCRIPTION></FIELD>
<FIELD ID="n_meas" datatype="int" name="n_meas" ucd="meta.number;obs">
   <DESCRIPTION>Number of measurements returned for this source</DESCRIPTION></FIELD>

[...]
```

MultiSpectrum: Vector columns (one entry per spectral point)

```
<FIELD ID="lambda" arraysize="*" datatype="float" name="lambda" ucd="em.wl" unit="um"
  utype="spec:Spectrum.Data.SpectralAxis.Value">
    <DESCRIPTION>SPHEREx wavelengths</DESCRIPTION></FIELD>
<FIELD ID="flux" arraysize="*" datatype="float" name="flux" ucd="phot.flux;meta.main" unit="uJy”
  utype="spec:Spectrum.Data.FluxAxis.Value">
    <DESCRIPTION>SPHEREx observed flux</DESCRIPTION></FIELD>
<FIELD ID="flux_err" arraysize="*" datatype="float" name="flux_err" ucd="stat.error;phot.flux;meta.main" unit="uJy"
  utype="spec:Spectrum.Data.FluxAxis.Accuracy.StatError">
    <DESCRIPTION>SPHEREx flux error best estimate</DESCRIPTION></FIELD>
<FIELD ID="lambda_width" arraysize="*" datatype="float" name="lambda_width" ucd="em.wl;instr.bandwidth" unit="um"
  utype="spec:Spectrum.Data.SpectralAxis.Accuracy.BinSize">
    <DESCRIPTION>SPHEREx wavelength widths</DESCRIPTION></FIELD>
[…]
<FIELD ID="obs_publisher_did" arraysize="46x*" datatype="char" name="obs_publisher_did" ucd="meta.ref.ivoid">
    <DESCRIPTION>Publisher dataset identifier in IRSA archive</DESCRIPTION></FIELD>
```

This is the spectrum itself!

Note the use of completely standard Utypes from SpectrumDM

MultiSpectrum: Defining the columns of the actual spectra

```
<GROUP ID="photometry" utype="ipac:Spectrum.ArrayData">
  <GROUP utype="spec:Spectrum.Data.FluxAxis">
   <FIELDref ref="flux" />
   <GROUP utype="spec:Spectrum.Data.FluxAxis.Accuracy">
    <FIELDref ref="flux_err" /></GROUP>
  </GROUP>
  <GROUP utype="spec:Spectrum.Data.SpectralAxis">
   <FIELDref ref="lambda" />
   <GROUP utype="spec:Spectrum.Data.SpectralAxis.Accuracy">
    <FIELDref ref="lambda_width" /></GROUP>
  </GROUP>
  <FIELDref ref="mjd" />
  <FIELDref ref="flux_bkg" />
  <FIELDref ref="flags" />
  [. . .]
  <FIELDref ref="obs_publisher_did" />
</GROUP>
```

[...]

MultiSpectrum: Defining a link to a cutout service

```
<RESOURCE type="meta" utype="ipac:MultiSpectrum-service">
 <DESCRIPTION>SPHEREx links service</DESCRIPTION>
 <GROUP name="inputParams">
  <PARAM name="ID" datatype="char" arraysize="*" ref="obs_publisher_did" value="">
   <DESCRIPTION>IVOA identifier of SPHEREx spectra image</DESCRIPTION></PARAM>
  <PARAM name="show_calib" datatype="int" arraysize="1" value="0"/>
 </GROUP>
 <PARAM name="standardID" datatype="char" arraysize="*" value="ivo://ivoa.net/std/DataLink#links-1.1"/>
 <PARAM name="accessURL" datatype="char" arraysize="*" value="https://irsa.ipac.caltech.edu/datalink/links/spherex"/>
</RESOURCE>

[...]
```

Cutout service is one of the responses from the DataLink links service

MultiSpectrum: Data

Wavelength:

<TD>0.7422174 0.7479611 0.7596464 0.76539785 0.77126 0.77133393 0.7774402 0.7895884 0.789796 0.7957131 0.8009711 0.8011317 0. 8082586 0.8084907 0.81465286 0.8198309 0.82007414 0.8274738 0.8275619 0.8341297 0.8392631 0.839369

[…]

4.720332 4.7306356 4.7448616 4.75482 4.764791 4.7792444 4.788908 4.796948 4.8131933 4.823305 4.831962 4.8485165 4.8589587 4.884129 4.893351 4.902525 4.9191003 4.9299383 4.9382176 4.955555 4.9663863 4.9667306 5.0026336</TD>

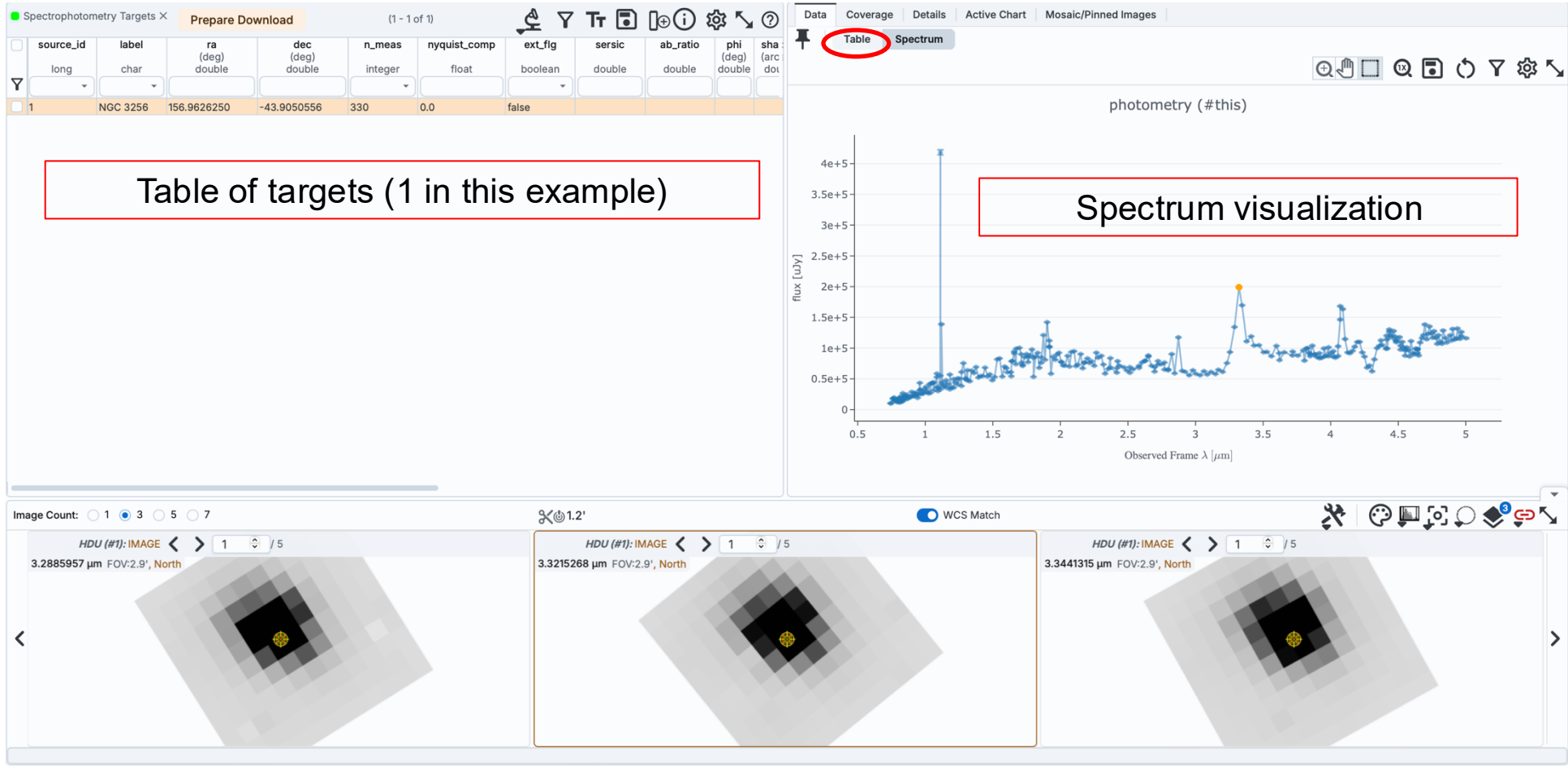
Flux:

<TD>10320.091 10312.704 17831.629 18624.732 14677.887 19507.969 15906.645 12358.775 13773.25 14517.548 17356.709 16384.41 12208.495 11803.636 19921 21439.32 13585.971 12648.5625 15201.918 26872.588 20374.516 15589.259 20815.492

[…]

127598.734 116954.305 106976.87 121005.44 110191.51 107046.914 128697.68 116371.56 109152 120869.31 112359.055 131179.55 114160.875 116092.86 131890.45 115173.945 126317.64 119571.51 116090.23</TD>

… all enabling this visualization in Firefly



Next steps – is MultiSpectrum something we can standardize?

- Or is there something else that's close enough to maturity to use?

Data Access (SSAP, and TAP+DataLink)

- Query for individual spectra (a relatively small number)
  - We plan to offer SSAP, with the URLs returned yielding single-spectrum SpectrumDM 1.2 VOTables
    - We still need to figure out how to provide access to both the “native” and the binned spectra.
    - We could return both as two <TABLE>s in a single <VOTable>,
      BUT this may not be great for sources in the deep fields, for which a) there may be two binned spectra available (all-sky depth, full-data depth), and b) the “native” spectrum may be much larger than a user would like to get automatically (20,000 data points – 200x the 102 points of the binned spectrum)
    - A better solution might be returning a DataLink “links service” URL as the Access.Reference, though this is strictly speaking an anachronism if used with SSAP v1.1.
    - Finally, the SSAP Association mechanism could be used to document that multiple rows in the SSAP response arise from the same source: native spectral data, all-sky-depth binned data, (optionally) full-depth binned data
  - We will probably also offer TAP access to a catalog of the sources for which spectra are available, with a service descriptor on the result table providing the path for retrieving the spectra for a source
    - Most likely the service descriptor would construct a URL for a DataLink “links service”
  - Neither of these really scale to queries for very large numbers of spectra: iteration over single retrievals

Data Access (HEALPix tiles via ObsTAP and DAP)

- Retrieval for bulk spectra in their native spatial sharding
  - Our Parquet spectral tiles for HEALPix order-six regions can be represented in ObsCore and CAOM2, with polygon approximations to the pixel shapes as s_region values
  - Data can be queried via ObsTAP, with dataproduct_type = spectrum, or measurements
  - For a “simple”-style query-parameter interface, we would have to use DAP (so let’s push that out!)

  - The results would be in our “MultiSpectrum” data model described above, ideally on the way to being standardized

- What about query-driven retrieval of large numbers of spectra, packaged dynamically?
  - Use cases: User-uploaded lists of object IDs? Selection by gross properties (e.g., z)?
  - At 10^4 – 10^5 – 10^6 + levels, iteration over individual spectra is awkward, both for retrieval and analysis
  - What would a standardized bulk-spectra-query API look like?
  - What’s the back end? A database? Files indexed by a database? HATS?

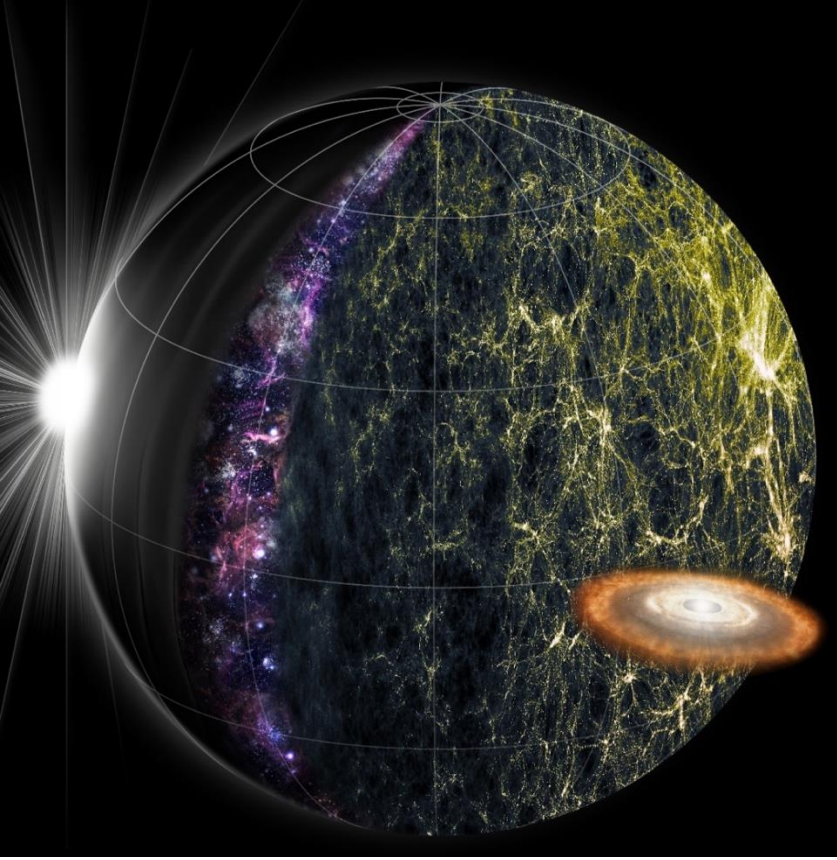
In-situ bulk processing of very large spectral catalogs – HATS/LSDB

- We anticipate a lot of interest in analyzing SPHEREx spectra alongside other large wide-area databases, both photometric and spectral (Rubin, Euclid, Roman, DESI, …)
- We've started discussions with the HATS/LSDB team at LINCC Frameworks on the application of their tools to this problem
  - HATS sharding: source-density-adaptive, so not all tiles would be at a uniform order 6
  - Array-valued data in rows: already part of the HATS landscape for time series, with nested-Pandas for analysis
  - Expecting an update on this today from the LINCC Frameworks team
- Each shard file would be a VOParquet file in the MultiSpectrum data model

Miscellanea

- VODML + MIVOT + SpectrumDM?
  - We haven't had the cycles to determine whether there's an alternative data model path ready to follow using this tooling. Realistically that's not going to be within our capabilities in the near term, but it would be good to at least check in on this at a high level.
- Reboot of a SpectrumDM v2+ effort?
  - Again, we're not adequately aware of where that stands

- Astropy ecosystem integration
  - We would like to encourage the development of tooling allowing the Astropy spectral-analysis ecosystem to work with IVOA standards (e.g., work with SpectrumDM in `specutils`)
    - Also integrate PyVO SSA support with `specutils` classes as the in-memory format
  - Bulk spectra: there's a SpectrumList class in `specutils` - can we make it work with our Parquet MultiSpectrum files?



SPHEREx

Jet Propulsion Laboratory
California Institute of Technology

Appendix: The All-Sky Cube and the HiPS standard(s)

All-Sky Cube contents

- We will produce a surface-brightness-calibrated map of the sky in each of our 102 reference spectral channels (reminder: 0.75-5.0um, R ~34-128)

- These maps are intended for general visualization and for the understanding of large-scale structure, NOT for point-source photometry
  - We currently plan to use the HEALPixel grid for the map tiles
- Pixel scale for initial release is TBD but will likely be between 9” and 13” (native pixels ~6.15”)
  - Standard HEALPixel scales near this are ~6.4” and ~12.9”

- The channel spacing is regular (logarithmic) within each of the six detectors, but varies discontinuously between detectors

- IRSA will provide all-sky pan-and-zoom functionality
  - Spectral extraction (“skewer”) from the cube is a requirement *(but these are not source spectra!)*

HiPS and the All-Sky Cube

- Native data product is “already HEALPix” so it is tempting to try to shoehorn it into HiPS

- HiPS 1.0 provides for a “cube” data format
  - Already supported by Firefly, Aladin, etc.

- Issues with this format for us:
  - Channel spacing is constrained to be regular in the third coordinate (via data_cube_cdelt3)
    - Wholly unsuitable for all 102 channels in a single data structure; inaccurate even within a detector’s band
    - We need a lookup-table solution for reporting the channel centers *and bandwidths*
  - Allocation of the files at Order 6 to directories produces ~1M files per directory (10,000 x 102)
    - Unsuitable for deployment as a static tree in a Posix filesystem on the back end but:
      can be mitigated by serving the HiPS URLs via a service that remaps the URLs to files in smaller directories
  - Distribution of the channels across files has performance implications for the “skewer” extraction
    - We would prefer the highest-resolution layer to be in FITS cubes with a lookup-table WCS for the 3rd axis
  - We wish to release additional per-pixel data (variance, flags) for the highest-resolution layer

Pre-HiPS-2.0 notions we've considered

- Devise a lookup-table extension to HiPS for the third coordinate
  - Envisioning a JSON data structure allowing the specification of more than one named value per plane: channel center and bandwidth
  - Uses a `properties` keyword not already reserved, so that non-SPHEREx-aware clients can still page through the planes (albeit without the ability to see wavelength values)

- Devise a mechanism for allowing FITS data for the highest-resolution layer to be in the form of cubes, not 102 separate files per tile
  - Provide the fully-quantitative FITS data only for the highest-resolution layer; webp/jpg/png for all layers
  - Again, make this "invisible" to non-SPHEREx-aware clients, BUT co-located with the HiPS tree
  - Use "HiPS Progenitors" to associate it with the HiPS-1.0-standard data, allowing tools like Aladin to still retrieve and directly access the individual FITS cubes

- Provide access to the FITS-cube data via ObsCore services in addition to via the "HiPS+"

Very preliminary thoughts about HiPS 2.0 (a few hours' thought in a busy week)

- SPHEREx finds the following things essential:
  - Preserve access to quantitative information about the 102 channels' definitions
  - Preserve the 102-channel view even when zoomed out spatially

  - Preserving access to the additional per-pixel data could be done "off to the side" and is less essential

- I've just started digesting the way the spectral dimension works in HiPS 2.0
  - The variation in R in the SPHEREx channels is about 4x – two orders on the binned frequency axis
    - It's not clear on a very quick read how to deal with this cleanly in a HiPS 2.0 representation without either duplicating
  - The project very specifically controls the channel definitions
    - We cannot "round off" to nearby FMOC-ized bins in the quantitative data we release
  - It's not clear how a HiPS 2.0 rendering would preserve some knowledge of this structure
  - I worry that the "spectral skewer" produced from a HiPS 2.0 would be misleading about the channel structure and resolution, but maybe I haven't read carefully enough

- We would love to find a way to release the data natively in a format like this so… we should talk!