Combining Data and Metadata: Hybrid Tabular File Formats

Mark Taylor   University of Bristol, UK   m.b.taylor@bristol.ac.uk

Introduction

Many considerations go into selecting a suitable file format for a given purpose when working with astronomy data. These include

- availability of compatible software
- ease of use by machines
- ease of use by humans
- efficiency of processing in one or several modes (sequential, random, parallel)
- storage size
- flexibility
- support for required data types (integer, floating, string, array, ...)
- ability to represent rich metadata (data semantics, provenance, ...)

Ideally, one existing file format presents itself as satisfying all the required criteria. Where that is not the case however, compromises may be made on one or several requirements, or the decision may be taken to invent a new format or adapt an existing one.

We discuss here the development and use of **hybrid data/metadata formats** for tabular data, in which the bulk data is stored in one (data-efficient but metadata-poor) table format, and associated semantic metadata is stored in some other (metadata-rich) format, with some arrangement to associate the data and metadata parts together.

We outline design considerations for such hybrid formats, especially how the data and metadata parts are associated, and list some examples of how this has been done.

Design Considerations

Questions to consider when designing a hybrid data/metadata format include:

- What software will be able to read/write it?
- Can it be read by standard software that understands the component data/metadata formats?
- Is there good compatibility between the data and metadata formats?
- How will the data/metadata association work?
- Is the data format fit for purpose?
- Is the metadata format fit for purpose?
- Can the metadata be easily viewed? Edited?

Compatibility of the formats is particularly important. If some columns in the data format cannot be described by the metadata format, **data/metadata mismatches** are possible. ❗

Association Options

Having identified one format to encode data and another to encode metadata, some means must be defined to join the two into a single unit that can be handled by I/O software.

This is usually done by adapting one of the formats to contain or reference an instance of the other. It can be done either way round:

Data wraps metadata ⧗̄:
An instance of the hybrid format is or resembles an instance of the data format, but with some way to associate an instance of the metadata format

Metadata wraps data ⧗̱:
An instance of the hybrid format is or resembles an instance of the metadata format, but with some way to associate an instance of the data format

If done carefully, this allows instances of the hybrid format to be processed by **non-hybrid-aware software** that understands the "wrapping" format without awareness of the "wrapped" metadata/data. ⚽

This wrapping can be done by **embedding** or **reference**:

Embedded ✉:
The content of the wrapped item is stored within the wrapper file, ideally in some way that does not disrupt wrapper file parsing

Referenced ☞:
The content of the wrapped item is stored in a file or resource separate to the wrapper file, using a pointer such as a URL or filename; this pointer may be stored within the wrapper file or maintained elsewhere

The **Referenced** option provides the option of manipulating the metadata separately from the data, but incurs the responsibility of keeping the two files or resources together which can be problematic. In most cases the **Embedded** option is more convenient and robust.

Format Example: VOParquet

The **VOParquet** hybrid format combines **Parquet** encoding of bulk data with **VOTable** encoding of semantic metadata, using the **Data wraps metadata** ⧗̄ model with **Embedded** ✉ association.
A VOParquet file is a **perfectly legal Parquet file** ⚽, encoding data as defined by the Parquet standard. But it also contains in the standard Parquet key-value metadata area a VOTable header encoding rich associated metadata. This VOTable header is used by VOParquet-aware software to decorate the parquet bulk data. Normal Parquet handling software that is unaware of the VOParquet convention however will simply ignore the extra metadata and process the Parquet data as usual. Since some Parquet columns cannot be described by VOTable metadata, **mismatches are possible** ❗. There is software support for the hybrid format in **TOPCAT** 🐈 and **Python** 🐍.

Some Known Hybrid Table Formats

		Metadata	
		VOTable	YAML
Data	Parquet	VOParquet⧗̄✉⚽❗📄🐍🐈 votable.parquet⧗̱☞❗🐍	MAML⧗̄✉⚽📄🐍
	FITS	FITS-plus⧗̄✉⚽🐈 FITS-serialized VOTable⧗̱☞✉📄🐈	
	CSV		ECSV✉📄🐍🐈

Data Formats

Parquet: Modern industry standard format for large/huge tabular datasets. Efficient storage/processing, many off-the-shelf packages; extremely limited standard semantic metadata, but key-value list allows custom metadata.

FITS: Flexible Image Transport System, venerable astronomy format for tables and arrays. Ubiquitous in astronomy, lean, efficient, easy to implement, but clunky, old-fashioned, not suited to parallel processing, restrictive metadata arrangements.

CSV: Comma-Separated Values, text-based table format. Read-write anywhere but inefficient and no metadata beyond column name.

Metadata Formats

VOTable: XML-based format for tables in astronomy, developed as an IVOA standard since 2003. Annotates column names, units, UCDs, DataLink service descriptors, space/time coordinate system information etc.

YAML: Yet Another Markup Language, human/machine-readable format for structured data/metadata. No implicit semantics.

Hybrid Formats

VOParquet: Parquet file with VOTable header in footer key-value list. Defined by IVOA Note Jan 2025. Being adopted by LSST, Gaia (DR4 downloads), HATS, CDS, others?

votable.parquet: VOTable-like file using external Parquet file for bulk data. I/O option of `astropy.io.votable` since Astropy 6.0 (2023); not usable elsewhere?

FITS-plus: FITS file with VOTable in primary HDU; since FITS and VOTable share datatypes, data/metadata match is good. Default FITS output from TOPCAT since 2004.

FITS-serialized VOTable: VOTable with data in embedded or referenced FITS file. Defined as part of VOTable standard since inception, but rarely used.

MAML: Parquet file with YAML metadata in footer key-value list. VOParquet-like for people who prefer YAML to VOTable; YAML block may also define non-Parquet data. Defined at `https://github.com/asgr/MAML-Format`; Being developed by WAVES/Data Central.

ECSV: CSV-like file with YAML header in `#`-comment lines. Defined in Astropy APE6. Used for Gaia DR3 downloads.

Legend

⧗̄	Data wraps metadata
⧗̱	Metadata wraps data
✉	Embedded wrapping
☞	Referenced wrapping
⚽	Readable by non-hybrid-aware software
❗	Data/metadata mismatch possible
📄	Published standard
🐍	Python supports I/O
🐈	TOPCAT supports I/O

ADASS XXXV, Görlitz, Germany, November 2025   Poster submission #196   `$Id: poster.tex,v 1.11 2025/10/17 14:22:45 mbt Exp $`