



HAL
open science

Time-frequency catalogue: JSON implementation and python library

Baptiste Cecconi, Corentin K Louis, Xavier Bonnin, Alan Loh, Mark B Taylor

► **To cite this version:**

Baptiste Cecconi, Corentin K Louis, Xavier Bonnin, Alan Loh, Mark B Taylor. Time-frequency catalogue: JSON implementation and python library. *Frontiers in Astronomy and Space Sciences*, 2023, 9, pp.1049677. 10.3389/fspas.2022.1049677 . hal-03983124

HAL Id: hal-03983124

<https://hal.science/hal-03983124>

Submitted on 10 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



OPEN ACCESS

EDITED BY

K.-Michael Aye,
Freie Universität Berlin, Germany

REVIEWED BY

Pablo Gutiérrez-Marqués,
Max Planck Institute for Solar System
Research, Germany
Hamood Ur Rehman,
University of Okara, Pakistan

*CORRESPONDENCE

Baptiste Cecconi,
baptiste.cecconi@observatoiredeparis.psl.eu

SPECIALTY SECTION

This article was submitted to Space
Physics, a section of the journal *Frontiers
in Astronomy and Space Sciences*

RECEIVED 20 September 2022

ACCEPTED 07 November 2022

PUBLISHED 09 February 2023

CITATION

Cecconi B, Louis CK, Bonnini X, Loh A and
Taylor MB (2023), Time-frequency
catalogue: JSON implementation and
python library.
Front. Astron. Space Sci. 9:1049677.
doi: 10.3389/fspas.2022.1049677

COPYRIGHT

© 2023 Cecconi, Louis, Bonnini, Loh and
Taylor. This is an open-access article
distributed under the terms of the [Creative
Commons Attribution License \(CC BY\)](#). The
use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

Time-frequency catalogue: JSON implementation and python library

Baptiste Cecconi^{1*}, Corentin K. Louis², Xavier Bonnini¹,
Alan Loh¹ and Mark B. Taylor³

¹LESIA, Observatoire de Paris, Université PSL, Sorbonne Université, Université Paris Cité, CNRS, Meudon, France, ²School of Cosmic Physics, DIAS Dunsink Observatory, Dublin Institute for Advanced Studies, Dublin, Ireland, ³H H Wills Physics Laboratory, University of Bristol, Bristol, United Kingdom

TFCat (Time-Frequency Catalogue) is a data interchange format based on JSON (JavaScript Object Notation), which has been initially designed for exchanging low frequency radio events and features. It defines several types of JSON objects and how they are combined to represent data related to temporal-spectral features of a time spectrogram (a.k.a., dynamic spectrum), their properties, and their temporal and spectral extents. This implementation is inheriting from the GeoJSON file format. The TFCat python library is implementing this specification and provides a software interface permitting to create, update and validate TFCat objects efficiently.

KEYWORDS

radioastronomy, data format, python library, catalog, JSON

1 Introduction

At the lower end of the radio frequency domain, i.e., below ≈ 100 MHz, the electromagnetic signatures are tracers of energetic and unstable particle populations rather than atomic and molecular transitions. Such features are usually identified with their spectral-temporal (also referred to as “time-frequency”) shape in time varying spectrograms (also referred to as dynamic spectra). Such time-frequency features are related to Earth and planetary magnetospheres (magnetospheric and auroral radio emissions), Earth and planetary lightnings (electrostatic discharges), Solar activity (solar radio bursts), as well as astrophysical radio sources such as Pulsars or Fast Radio Bursts. For a review of the low frequency planetary radio emissions, see [Zarka \(2004\)](#). **Figure 1** shows an example dynamic spectrum recorded by the Cassini/RPWS (Radio and Plasma Waves Science) instrument ([Gurnett et al., 2004](#)) displaying Solar and Terrestrial low frequency radio features.

Since the early 1960's, a corpus of catalogues identifying and documenting such Jovian radio emissions ([Barrow and Desch, 1980](#); [Barrow, 1981](#); [Leblanc et al., 1981](#), [Leblanc et al., 1983](#); [Aubier and Genova, 1985](#); [Leblanc et al., 1989](#), [Leblanc et al., 1990](#); [Riihimaa, 1991](#); [Leblanc et al., 1993](#); [Marques et al., 2017](#)), and solar radio bursts ([Maxwell et al., 1963](#); [Bernold, 1980](#); [Kayser et al., 1988](#); [Zhang et al., 2018](#)) have been

published, or are available online (such as the Bleien Solar Radio Data legacy web page¹), providing access to: lists of event bounding boxes in the temporal-spectral plane; more refined contour shapes; or linear features. However, there was no standard exchange format, leading to difficulties in sharing and reusing such catalogues.

The TFCat (Time-Frequency Catalogue) specification has been developed with the aim of facilitating the recording, rese and processing of low frequency radio temporal-spectral events. In the past two years, several catalogues have been published using the TFCat format, and tools are now implementing interfaces on top of this specification, for labelling observations, displaying catalogues, or even using the catalogue shapes to select and process observational data (Cecconi et al., 2021a; Fogg et al., 2021; Louis et al., 2021; Taubenschuss et al., 2021; Wu et al., 2021, Wu et al., 2022b, Wu et al., 2022a).

In this paper, we will present the needs and use cases identified in the preliminary stages of the TFCat specification project. We then detail the proposed catalogue model and its implementation. The associated python library is also presented, with a series of examples and applications.

2 Needs and use cases

The MASER service (Measuring, Analysing and Simulating Emissions in the Radio range) is a science ready toolkit for low frequency radio astronomy (Cecconi et al., 2020). During its inception, the storing, processing and sharing of labelled features in radio frequency dynamic spectra has been identified as one of the major community needs. The main complexity is the shape representation of the features, as a series of coordinates in the temporal-spectral plane. The MASER team has thus explored various applicable solutions based on existing tools and standards in the astronomy, heliophysics and Earth sciences.

The astronomy community has developed an interoperability framework called the “Virtual Observatory” and maintained by the IVOA (International Virtual Observatory Alliance, <https://ivoa.net>). They define many standards allowing to build complex interoperable data infrastructures. For the purpose of catalogues and events, three standards are applicable: the VOTable (Virtual Observatory Table) format (Ochsenbein et al., 2019), the VOEvent (Virtual Observatory Event) model (Seaman et al., 2011) and the STC (Space-Time Coordinate) model (Rots, 2007) or its upcoming successor². Those formats and models are very well designed for

astronomical catalogues and events, including time intervals, spectral ranges, spatial regions on the sky (most often using Right Ascension (RA) and Declination (Dec) coordinates). However, there is currently no way to build and share more detailed geometries mixing temporal and spectral coordinates using those tools. The heliophysics community has also defined a set of standards including a heliophysics event list (HPEventList³), which is using CSV (comma separated values) and VOTable as serialization formats. The same restriction applies as for the astronomy applications.

The description of two-dimensional (2D) geometries or coverage information can be done using various tools and methods. The solar physics community has been using the “chaincodes” method (see, e.g., the *SunPy* implementation⁴), which describes a 2D geometry specifying a starting point on an image and then listing the pixel per pixel path along the feature contour. Such implementation is computationally very efficient, but is providing geometry in pixel coordinates, requiring the access to the original data (and metadata) to reuse the described feature. In the astronomy framework, the MOC (HEALPix –Hierarchical Equal Area isoLatitude Pixelisation– Multi-Order Coverage map) standard (Fernique et al., 2019) is currently allowing to describe spatial and temporal complex coverage maps, however the spectral dimension is not considered yet in the standard. Finally in the Earth science community, feature geometries are using the GeoJSON format (Butler et al., 2016). This format is using geographic coordinates to describe features and proposes several classes of features (such as points, segments or polygons).

In order to facilitate the adoption by catalogue providers (e.g., authors of publications), the selected solution must be easy to implement and should avoid dependencies on too specific software. On the user side, the same statement applies with the other requirements that it should be easy for tool and science platform developers to implement the solution.

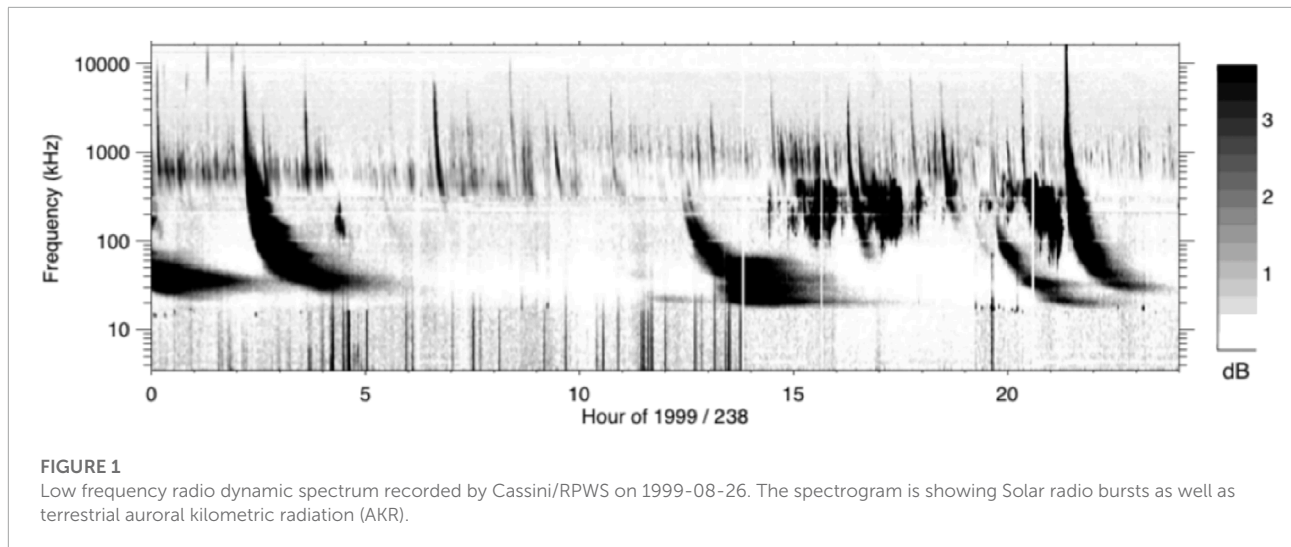
Besides the description of the temporal-spectral feature geometries, it shall be possible to set additional properties for each feature (e.g., a label, an identifier linking to the original observation, or any property or statistical quantity related to the observed feature). Such properties shall be defined properly, so that users can reuse and process them. Furthermore, the catalogue shall contain metadata covering the scope of the catalogue itself (such as authors, references, a title). Metadata on the source datasets, from which the feature catalogue is derived, should also be referenced. Finally the temporal and spectral axes must be properly defined within the catalogue.

1 <http://soleil4ds.ch/solarradio/>.

2 Astronomical Coordinates and Coordinate Systems (2021) Proposed IVOA Recommendation. Version 1.0. <https://www.ivoa.net/documents/Coords/20210924/index.html>.

3 Heliophysics Event List: <https://spase-group.org/docs/conventions/HDMC-Event-List-Specification-v1.0.4.pdf>.

4 SunPy.Chaincode: <https://docs.sunpy.org/en/stable/generated/api/sunpy.net.helio.Chaincode.html>.



3 Proposed catalogue model

The TFCat model provides a way to describe a series of features referenced by their coordinates (or set of coordinates) in the temporal-spectral plane, as observed from a radio instrument (a ground based telescope or onboard a spacecraft), and identified by an agent (a scientist or a software). Each feature of the catalogue is defined by a set of temporal and spectral coordinates, and a geometry type. Features can have additional properties, depending on the science case, which content shall be defined in the catalogue. The temporal and spectral coordinate frame in use in the catalogue shall be defined uniquely in a catalogue file, or identified by unique coordinate system name. The catalogue can have general properties (e.g., header type metadata). The TFCat base model elements are:

- *TFCat Geometry*—There are 7 types of TFCat Geometry: *Point*, *MultiPoint*, *LineString*, *MultiLineString*, *Polygon*, *MultiPolygon*, with an additional *GeometryCollection*, which is a set of any of the six other geometry types. Each geometry is composed of a set of coordinate pairs. A *LineString* is defined by a list of connected *Point* features, forming a continuous path. A *Polygon* is defined by a closed list of connected *Point* features, forming a closed contour. The feature type definitions are inspired from the GeoJSON specification (Butler et al., 2016).
- *TFCat Feature*—A TFCat Feature contains one or several TFCat Geometries, and a set of properties, provided as a set of (key, value) pairs.
- *TFCat Field*—A TFCat Field is defining the TFCat feature properties. Its minimal content includes a name, a description, a data type and a UCD (Unified Content Descriptors) (Cecconi et al., 2021b). It may also include a unit [following the VOUnit –Virtual Observatory

Unit–specification (Derriere et al., 2014)], or a set of allowed values, when applicable.

- *TFCat Property*—A TFCat Property is a (key, value) pair. There are two types of TFCat Properties:
 - *TFCat Feature Property*—TFCat feature properties are used in a TFCat Feature. They must be defined with a corresponding TFCat Field.
 - *TFCat Collection Property*—TFCat collection properties are used in a TFCat Feature Collection. They should provide generic information on the catalogue, such as: title, authors, reference, instrument, etc.
- *TFCat Coordinate Reference System*—A TFCat Coordination Reference System (CRS) contains a description of the time axis, a description of the spectral axis, and a reference position. The TFCat CRS could be of three types: *name*, *link* or *local*.

3.1 Javascript object notation implementation

The TFCat model has been implemented using the JSON format (Bray, 2014), reusing the structure of the GeoJSON specification. The TFCat JSON specification is available (Cecconi et al., 2022b). The specification is maintained on a git repository⁵. When applicable the astronomy standards have been implemented (UCD (Preite Martinez et al., 2018), VOUnit (Derriere et al., 2014), and list of known keywords, like, e.g., reference positions).

⁵ TFCat specification: <https://gitlab.obspm.fr/maser/catalogues/catalogue-format>.

The TFCat CRS specification aligns on GeoJSON CRS definition. Currently, the TFCat CRS must be defined in the TFCat object instance. However, a set of predefined time axis are available within the specification (e.g., *unix*, *jd*, *mjd*, etc).

3.2 Catalogue validation

A JSON Schema document⁶ has been prepared to validate the catalogues. However, the JSON schema only covers part of the specification, since some constraints could not be implemented therein (like, e.g., the need to have a TFCat Field defined for each TFCat Feature Property). An example of TFCat validation using the JSON schema is provided in [Section 4.1](#).

A full validator (based on the TFCat specification) has been implemented in *STILTS*⁷ (Starlink Tables Infrastructure Library Tool Set, [Taylor, 2006](#)), version 3.4-7 or later.

This java-based tool can be called on the command line:

```
1 java -classpath stilts.jar uk.ac.starlink.tfcats.Validate <filename.json>
```

Listing 1: Validating a TFCat file

4 Time-frequency catalogue python library

Since the TFCat JSON implementation is closely inheriting from GeoJSON, many tools from the Earth science community can be reused. First, the TFCat python library⁸ has been developed starting from the code of a python GeoJSON library⁹. The TFCat library also makes use of the *shapely* library¹⁰, which can be used for managing and processing TFCat features in general.

4.1 Creating a time-frequency catalogue feature collection

The two main elements required to start preparing a TFCat feature collection is the choice of the CRS and the list of feature geometries (with its set of time and spectral coordinates). In

6 TFCat JSON Schema v1.0: <https://voparis-ns.obspm.fr/maser/tfcats/v1.0/schema#>.

7 STILTS website: <http://www.starlink.ac.uk/stilts/>.

8 TFCat Python library: <https://gitlab.obspm.fr/maser/catalogues/tfcats>.

9 GeoJSON: <https://github.com/jazzband/geojson>.

10 shapely: <https://shapely.readthedocs.io/en/stable/>.

the following code snippet, a CRS object is created with a time axis using the *unix* timestamp representation, a spectral axis (frequency in kHz) and a reference position set to the Juno spacecraft.

```
1 from tfcat import CRS
2 crs = CRS({
3     "type": "local",
4     "properties": {
5         "name": "Time-Frequency",
6         "time_coords_id": "unix",
7         "spectral_coords": {
8             "type": "frequency",
9             "unit": "kHz",
10        },
11     "ref_position_id": "juno",
12     }
13 })
```

Listing 2: Example CRS definition

In the following example, we create a TFCat feature with a TFCat *Polygon* geometry and two custom feature *properties* called *feature_type* and *integrated_flux_density*. The names and values of feature properties are not constrained by the TFCat specification. However, the specification requires to include TFCat Fields defining any TFCat feature property that may be included in a Feature. The set of coordinates are available in two variables called *tt* and *ff*, containing respectively the time coordinates (as a *unix* timestamp), and the frequency coordinates (in kHz). Note, that a *Polygon* geometry requires that the last coordinate point of the input sequence should be the same as the first one (closed polygon condition). It also requires that the segments forming the polygon are not crossing over (no cross-over condition). The polygon must also be curled counter-clockwise. We use the *shapely* package to check the latter condition.

```
1 from tfcat import Polygon, Feature
2 from shapely.geometry import LinearRing
3 # packing coordinates into a list
4 coords = list(zip(tt, ff))
5 # checking if list of coordinates is curled in
6   the direct trigonometric sense (counter-
7   clockwise)
8 if not LinearRing(coords).is_ccw:
9     coords = coords[::-1]
10 # creating the TFCat Feature
11 feature = Feature(
12     geometry=Polygon([coords]),
13     properties={
14         "feature_type": "Io-A",
15         "integrated_flux_density": 1.22e-19,
```

```

14     },
15 )

```

Listing 3: Creating a TFCat Polygon

Each *properties* must be defined at the main level of the feature collection object, using a set of TFCat Field objects, as shown below.

```

1 fields = {
2     "feature_type": {
3         "info": "Feature Type",
4         "datatype": "str",
5         "ucd": "meta.id",
6     },
7     "integrated_flux_density": {
8         "info": "Integrated Flux Density",
9         "datatype": "float",
10        "ucd": "phot.flux.density;stat.mean",
11        "unit": "W m-2 Hz-1",
12    }
13 }

```

Listing 4: Defining Fields

The feature collection can also include collection-level *properties*, listing global metadata for the collection, such as the author, the instrument, etc. The next code snippet shows a set of collection level properties. We recommend to use keywords compliant with the EPNcore (Europlanet core) metadata dictionary (Erard et al., 2022) for Solar System related collections. A *data_source_reference* property may be included to refer to the data collection used to derive the catalogue. The value of the data source reference should be a DOI if it is available for the progenitor resource. However, any type of well-known community reference could be used, such as, e.g., a SPASE (Roberts et al., 2018) resource identifier or a NASA/PDS (Planetary Data System) identifier¹¹. For authors, we recommend to use a list of dictionaries containing *GivenName*, *FamilyName*, *ORCID* and *Affiliation* (using ROR¹² when possible). The current version of the TFCat specification does not define mandatory properties, nor does it define the object type of collection level properties. This may be implemented in a future version, in order to improve the interoperability and reusability.

```

1 properties = {
2     "instrument_host_name": "Juno"
3     "instrument_name": "Waves",

```

¹¹ Citing PDS4 Data: <https://pds.nasa.gov/datastandards/citing/>.

¹² Research Organisation Registry: <https://ror.org>.

```

4     "title": "Catalogue of Juno Radio emissions"
5     ,
6     "authors": [
7         {
8             "GivenName": "Corentin K.",
9             "FamilyName": "Louis",
10            "ORCID": "https://orcid.org/0000-0002-9552-8822",
11            "Affiliation": "https://ror.org/051sx6d27"
12        },
13    ],
14    "target_name": "Jupiter",
15    "target_class": "planet",
16    "target_region": "magnetosphere",
17    "feature_name": "radio emissions",
18    "bib_reference": "10.1029/2021JA029435",
19    "data_source_ref": "http://doi.org/10.25935/6jg4-mk86",
20    "publisher": "PADC/MASER",
21    "version": "1.0",

```

Listing 5: Defining collection level properties

Finally, the collection can be built, assuming a list of features is available in an object called *features*. The JSON schema URI can also be referenced. The default value is set to the latest released TFCat JSON schema URI (i.e., <https://voparis-ns.obspm.fr/maser/tfcats/v1.0/schema#> at the time of writing).

```

1 from tfcat import FeatureCollection
2 SCHEMA_URI = "https://voparis-ns.obspm.fr/maser/tfcats/v1.0/schema#"
3 # creating the collection
4 collection = FeatureCollection(
5     schema=SCHEMA_URI,
6     features=features,
7     properties=properties,
8     fields=fields,
9     crs=crs,
10 )

```

Listing 6: Finalizing a TFCat collection

The TFCat collection can then be checked against the JSON Schema.

```

1 from tfcat.validate import validate
2 validate(collection)

```

Listing 7: Validating a TFCat collection

Writing the collection into a file is straightforward, since the TFCat module implements the same interfaces as the *json* python module for file system interactions (i.e., *dump()*, *dumps()*, *load()*, *loads()* methods):

```
1 # write out the collection on disk
2 from tfcat import dump
3 filename = "tfcat.json"
4 with open(filename, "w") as f:
5     dump(collection, f)
```

Listing 8: Dumping into File

4.2 Loading a time-frequency catalogue feature collection

The TFCat module also provides capabilities to load TFCat JSON files from the local file system, or from a remote location:

```
1 from tfcat import TFCat
2 local_cat = TFCat.from_file(<filepath>)
3 remote_cat = TFCat.from_url(<URL>)
```

Listing 9: Loading local or remote TFCat file

In the following example, we load TFCat file available in [Taubenschuss et al. \(2021\)](#). The script first displays the coordinate reference system (using the *crs* property), and the geometry type of the first feature. It then plots the two first features of the catalogue (see [Figure 2](#)), and then shows the use of the temporal and spectral converters, which can be used to load the time-frequency coordinates as an *astropy.time.Time* object for the time component, and an *astropy.units.Quantity* for the spectral component.

```
1 from tfcat import TFCat
2 tf_file = 'cassini_faraday_patches_2006.json'
3 cat = TFCat.from_file(tf_file)
4 cat.crs
5 # {"properties": {"name": "Time-Frequency", "
6   "ref_position_id": "cassini-orbiter",
7   "spectral_coords": {"type": "frequency", "unit": "Hz"}, "time_coords_id": "unix"},
8   "type": "local"}
9 cat.feature(0).geometry.type
10 # 'MultiPoint'
11 cat.plot_feature(0)
12 cat.plot_feature(1)
13 cat.crs.time_converter(cat.feature(0).tmin).isot
14 # '2006-09-12T09:02:26.750'
15 cat.crs.spectral_converter(cat.feature(0).fmax)
```

```
15 # <Quantity 43469. Hz>
```

Listing 10: Example script

4.3 Module installation and dependencies

The TFCat module is distributed under MIT licence. The module dependencies are limited to a small set of external modules: *astropy* ([Astropy Collaboration et al., 2022](#)), *matplotlib* ([Hunter, 2007](#)), *shapely* ([Gillies et al., 2007](#)) and *jsonschema* ([Berman et al., 2022](#)). It is available on PyPI¹³ and can thus be installed with pip:

```
1 pip install tfcat
```

Listing 11: Install TFCat with pip

5 Applications

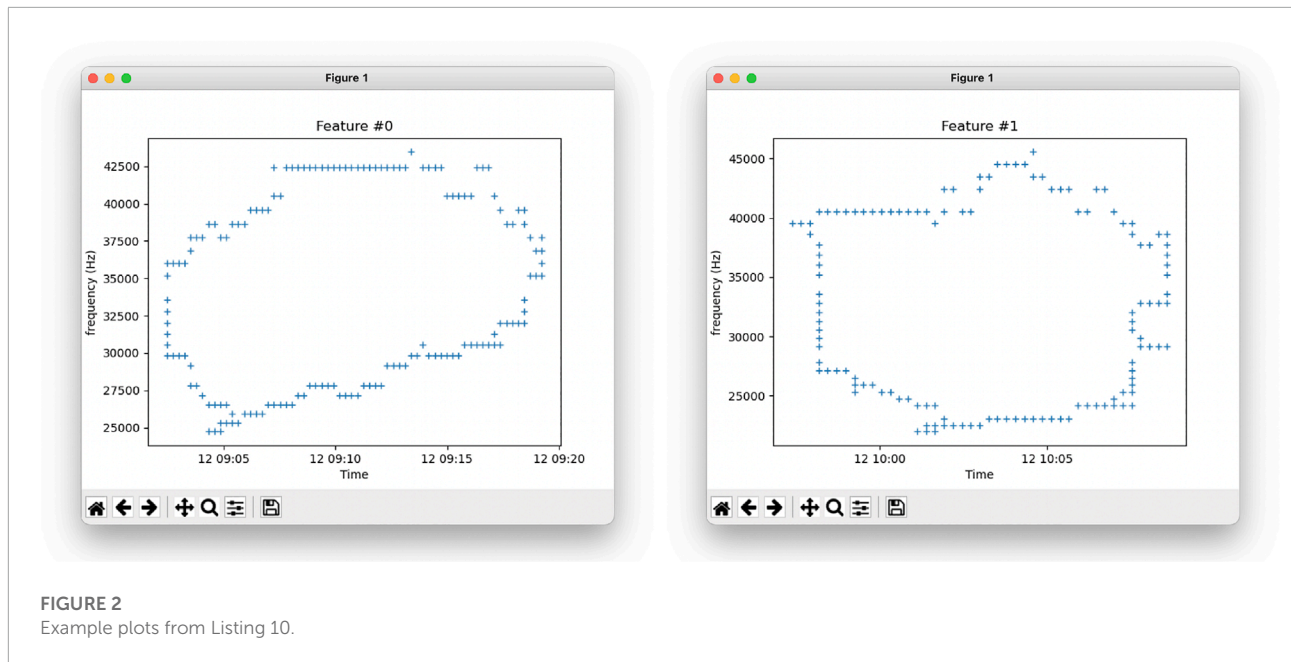
TFCat has already been used in several published data collections ([Cecconi et al., 2021b](#); [Fogg et al., 2021](#); [Louis et al., 2021](#); [Taubenschuss et al., 2021](#); [Smith et al., 2022](#); [Wu et al., 2021](#), [Wu et al., 2022b](#), [Wu et al., 2022a](#)). In order to showcase the capabilities of the TFCat format and its python implementation, 3 Jupyter notebook tutorials have been prepared, and are available in [Cecconi et al. \(2022a\)](#)¹⁴:

- *TFCat MultiPoint feature and feature processing* (MultiPoint_Example.ipynb) is showing how to use TFCat together with *shapely* to process a *MultiPoint* geometry into a *Polygon*.
- *TFCat MultiPolygon feature and data filtering example* (MultiPolygon_Example.ipynb) is showing how to use TFCat together with *shapely* and *das2* ([Piker et al., 2019](#)) to use a *MultiPolygon* geometry to select data and perform statistical analysis.
- *TFCat Multiple Catalogue Example* (Multiple_Catalogue_Example.ipynb) is showing how to compare two catalogues containing features derived from the same physical events.

The TFCat model and python library are used in the *SPectrogram Analysis and Cataloguing Environment* (SPACE) Labelling Tool ([Louis et al., 2022](#)). TFCat features can also be loaded and displayed in *Autoplot* ([Faden et al., 2010](#)). Full

¹³ <https://pypi.org/project/TFCat/>.

¹⁴ <https://doi.org/10.25935/qwpa-7t05>.



support is not currently available, but an example Jython script is provided¹⁵ by the Autoplot team.

6 Conclusion and future prospects

The TFCat format and its associated Python library are providing the low frequency radio astronomy community with a novel tool enabling interoperability and reusability of temporal-spectral features. The published catalogues are demonstrating its usefulness through various use cases. The proposed notebooks are showing its usability. Furthermore, the accessibility is guaranteed by the open access policies of the current TFCat catalogue publishers. The findability of the TFCat catalogues can be implemented using existing community data discovery interfaces (such as EPN-TAP (Erard et al., 2022) for Solar System sciences). In order to enable the findability of the features, an implementation using the TAP Table Access Protocol, (Dowler et al., 2019), is under study.

The TFCat format is now used in publications and tools to store, share and process temporal-spectra features. The GeoJSON inherited specification allows to reuse python based libraries (such as *shapely*) for processing the geometries. At Observatoire de Paris, the MASER team is integrating TFCat in their machine learning pipelines related to planetary and solar radio emissions, as well as in their radio data flagging pipelines (for excluding radio frequency interferences in the data processing).

¹⁵ Autoplot Jython script: <https://github.com/autoplot/scripts/tree/master/formats/tfcats>.

The TFCat collection properties are currently loosely defined: (a) the list of keywords is not defined in the TFCat specification; and (b) there is no metadata associated to them. This will be improved in a future major version, since this would require to refactor the already published collection. The specification foreseen for the TFCat collection properties would have the same definition as the TFCat Field items, with an additional *value* attribute, containing the value of the property. Such specification would improve TFCat reusability and interoperability, and it would make the format more compatible with other formats, such as VOTable.

Discussions are also ongoing with other communities dealing with temporal-spectral features, such as transient astrophysics (e.g., pulsars or fast radio bursts studies), meteor radio echoes, or even space based *in-situ* particle observations, for which the spectral axis would then represent an energy, mass or mass per charge axis.

Data availability statement

The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Author contributions

BC and CL developed the TFCat specification. BC developed the TFCat JSON Schema. BC, XB, and AL developed the TFCat python library. MT reviewed the TFCat specification, and developed the TFCat validator. BC wrote the first version

of the paper. XB, AL, and MT commented and proofread the manuscript.

Funding

The Europlanet 2024 Research Infrastructure (EPN2024RI) project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871149. CKL's work at DIAS is supported by the Science Foundation Ireland Grant 18/FRL/6199. BC and MT's contributions for this study are supported by the EPN2024RI project. BC, XB, and AL acknowledge additional funding provided in France by the Centre National d'Études Spatiales (CNES), and Action Spécifique Observatoire Virtuel (ASOV). Support from Paris Astronomical Data Centre (PADC) is acknowledged.

References

- Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., et al. (2022). The astropy project: Sustaining and growing a community-oriented open-source project and the latest major release (v5.0) of the core package. *Astrophys. J.* 935, 167. doi:10.3847/1538-4357/ac7c74
- Aubier, M. G., and Genova, F. (1985). A catalogue of the high frequency limit of the Jovian decameter emission observed by Voyager. *Astron. Astrophys. Suppl. Ser.* 61, 341–351.
- Barrow, C. H. (1981). A catalogue of Jupiter's decametric emission observed by Voyager-1 and by Voyager-2 in the range 15–40 MHz. *Astron. Astrophys. Suppl. Ser.* 46, 111–114.
- Barrow, C. H., and Desch, M. D. (1980). Non-Io decametric radiation from Jupiter at frequencies above 30 MHz. *Astron. Astrophys. Suppl. Ser.* 86, 339–341.
- Berman, J., Sterling, C., Taprest, R., Nezbeda, H., Chen, W., Rosen, S., et al. (2022). python-jsonschema/jsonschema: v4.16.0. Zenodo. doi:10.5281/zenodo.7064213
- Bernold, T. (1980). A catalogue of fine structures in type IV solar radio bursts. *Astron. Astrophys. Suppl. Ser.* 42, 43–58.
- Bray, T. (2014). The JavaScript object notation (JSON) data interchange format, RFC 7159. doi:10.17487/RFC7159
- Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., and Schaub, T. (2016). The GeoJSON format. *IETF RFC 7946*. doi:10.17487/RFC7946
- Cecconi, B., Bonnini, X., Loh, A., and Taylor, M. (2022a). TFCat (Time-Frequency catalogue): JSON implementation and Python library. Supplementary material. *PADC*. doi:10.25935/QWPA-7T05
- Cecconi, B., Debisschop, L., and Lamy, L. (2021a). A catalogue of Jovian decametric radio observations from January 1978 to December 1990 (Version 1.0) [Data set]. *PADC*. doi:10.25935/1FQM-FS07
- Cecconi, B., Loh, A., Sidaner, P. L., Savalle, R., Bonnini, X., Nguyen, Q. N., et al. (2020). Maser: A science ready toolbox for low frequency radio astronomy. *Data Sci. J.* 19, 1062. doi:10.5334/dsj-2020-012
- Cecconi, B., Louys, M., Preite Martinez, A., Derrière, S., Ochsenein, F., Erard, S., et al. (2021b). UCD1+ controlled vocabulary - updated list of terms version 1.4. *IVOA Endorsed Note*. doi:10.5479/ADS/bib/2021ivoa.spec.0616C
- Cecconi, B., Taylor, M. B., Bonnini, X., and Loh, A. (2022b). JSON implementation of time-frequency radio catalogues: TFCat (version 1.0). *PADC*. doi:10.25935/6068-8528
- Derrière, S., Gray, N., Demleitner, M., Louys, M., and Ochsenein, F. (2014). Units in the VO version 1.0. *IVOA Recomm.* doi:10.5479/ADS/bib/2014ivoa.spec.0523D
- Dowler, P., Rixon, G., Tody, D., and Demleitner, M. (2019). Table Access Protocol version 1.1. *IVOA Recomm.* doi:10.5479/ADS/bib/2019ivoa.spec.0927D
- Erard, S., Cecconi, B., Le Sidaner, P., Demleitner, M., and Taylor, M. B. (2022). Epn-tap: Publishing solar system data to the virtual observatory. *IVOA Recomm.* version 2.0. doi:10.5479/ADS/bib/2022ivoa.spec.0822E
- Faden, J. B., Weigel, R. S., Merka, J., and Friedel, R. H. W. (2010). Autoplot: A browser for scientific data on the web. *Earth Sci. Inf.* 3, 41–49. doi:10.1007/s12145-010-0049-0
- Fernique, P., Boch, T., Donaldson, T., Durand, D., O'Mullane, W., Reinecke, M., et al. (2019). Moc - HEALPix multi-order coverage map version 1.1. *IVOA Recomm.* doi:10.5479/ADS/bib/2019ivoa.spec.1007F
- Fogg, A. R., Jackman, C., Waters, J., Bonnini, X., Lamy, L., Cecconi, B., et al. (2021). Bursts of auroral kilometric radiation individually selected from wind/waves data. *PADC*. doi:10.25935/HFJX-XX26
- Gillies, S., van der Wel, C., Van den Bossche, J., Taves, M. W., Arnott, J., Ward, B. C., et al. (2007). Shapely (Version 2.0) [Computer software]. Available at <https://github.com/shapely/shapely>.
- Gurnett, D. A., Kurth, W. S., Kirchner, D. L., Hospodarsky, G. B., Averkamp, T. F., Zarka, P., et al. (2004). The cassini radio and plasma wave investigation. *Space Sci. Rev.* 114, 395–463. doi:10.1007/s11214-004-1434-0
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Comput. Sci. Eng.* 9, 90–95. doi:10.1109/MCSE.2007.55
- Kayser, S. E., Bougeret, J. L., Fainberg, J., and Stone, R. G. (1988). A catalog of interplanetary type III storms. *Astron. Astrophys. Suppl. Ser.* 73, 243–253.
- Leblanc, Y., de La Noe, J., Genova, F., Gerbault, A., and Lecacheux, A. (1981). A catalogue of Jovian decametric radio observations from January 1978 to December 1979. *Astron. Astrophys. Suppl. Ser.* 46, 135–149.
- Leblanc, Y., Gerbault, A., Denis, L., and Lecacheux, A. (1993). A catalogue of Jovian decametric radio observations from January 1988 to December 1990. *Astron. Astrophys. Suppl. Ser.* 98, 529–546.
- Leblanc, Y., Gerbault, A., and Lecacheux, A. (1989). A catalogue of Jovian decametric radio observations from January 1982 to December 1984. *Astron. Astrophys. Suppl. Ser.* 77, 425–438.
- Leblanc, Y., Gerbault, A., Lecacheux, A., and Boudjada, M. Y. (1990). A catalogue of Jovian decametric radio observations from January 1985 to December 1987. *Astron. Astrophys. Suppl. Ser.* 86, 191–207.
- Leblanc, Y., Gerbault, A., Rubio, M., and Genova, F. (1983). A catalogue of Jovian radio observations from January 1980 to December 1981. *Astron. Astrophys. Suppl. Ser.* 54, 135–148.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Louis, C. K., Jackman, C. M., Mangham, S. W., Smith, K. D., O'Dwyer, E. P., Empey, A., et al. (2022). The "spectrogram analysis and cataloguing environment" (space) labelling tool. *arXiv astro-ph.IM*. doi:10.3389/fspas.2022.1001166
- Louis, C. K., Zarka, P., and Cecconi, B. (2021). Catalogue of Jupiter radio emissions identified in the Juno/Waves observations (Version 1.0) [Data set], PADC. doi:10.25935/NHB2-WY29
- Marques, M. S., Zarka, P., Echer, E., Ryabov, V. B., Alves, M. V., Denis, L., et al. (2017). Statistical analysis of 26 yr of observations of decametric radio emissions from Jupiter. *Astron. Astrophys.* 604, A17. doi:10.1051/0004-6361/201630025
- Maxwell, A., Hughes, M. P., and Thompson, A. R. (1963). Catalog of type II(Slow-Drift) and type IV(continuum) solar radio bursts. *J. Geophys. Res.* 68, 1347–1354. doi:10.1029/JZ068i005p01347
- Ochsenbein, F., Taylor, M., Donaldson, T., Williams, R., Davenhall, C., Demleitner, M., et al. (2019). VOTable format definition version 1.4. *IVOA Recomm.* doi:10.5479/ADS/bib/2019ivoa.spec.10210
- Piker, C., Granroth, L., Mukherjee, J., Pisa, D., Cecconi, B., Kopf, A., et al. (2019). Lightweight federated data networks with das2 tools. *Earth Space Sci. Open Archive*. doi:10.1002/essoar.10500359.1
- Preite Martinez, A., Louys, M., Cecconi, B., Derriere, S., Ochsenbein, F., The Semantic IVOA working group, (2018). The UCD1+ controlled vocabulary Version 1.3. *IVOA Recomm.* doi:10.5479/ADS/bib/2018ivoa.spec.0527M
- Riihimaa, J. J. (1991). Evolution of the spectral fine structure of Jupiter's decametric S-storms. *Earth Moon Planets* 53, 157–182. doi:10.1007/BF00057430
- Roberts, D. A., Thieman, J., Génot, V., King, T., Gangloff, M., Perry, C., et al. (2018). The SPASE data model: A metadata standard for registering, finding, accessing, and using heliophysics data obtained from observations and modeling. *Space weather*. 16, 1899–1911. doi:10.1029/2018sw002038
- Rots, A. H. (2007). Space-time coordinate metadata for the virtual observatory version 1.33. *IVOA Recomm.* doi:10.5479/ADS/bib/2007ivoa.spec.1030R
- Seaman, R., Williams, R., Allan, A., Barthelmy, S., Bloom, J., Brewer, J., et al. (2011). Sky event reporting metadata version 2.0. *IVOA Recomm.* doi:10.5479/ADS/BIB/2011ivoa.spec.0111S
- Smith, K. D., Louis, C. K., Jackman, C. M., Fogg, A. R., O'Dwyer, E., Waters, J. E., et al. (2022). Catalogue of auroral kilometric radiations (AKR) observed by the polar spacecraft (1.2). Zenodo. doi:10.5281/zenodo.7260994
- Taubenschuss, U., Cecconi, B., and Lamy, L. (2021). Catalogue of faraday rotation patches identified in saturn kilometric radiation (SKR) observations by cassini/RPWS/HFR (Version 1.0) [Data set]. PADC. doi:10.25935/R11G-6J63
- Taylor, M. B. (2006). "Stilts - a package for command-line processing of tabular data." Editors C. Gabriel, C. Arviset, D. Ponz, and S. Enrique (San Francisco: Astronomical Society of the Pacific Conference Series), 351, 666. *Astronomical Data Analysis Softw. Syst.* XV
- Wu, S., Ye, S. Y., Fischer, G., Wang, J., Long, M. Y., Menietti, J. D., et al. (2021). Statistical study of spatial distribution and polarization for saturn narrowband emissions. Supplementary material: Event catalogue (Version 1.0) [Data set]. PADC. doi:10.25935/ZMMZ-CA38
- [Dataset] Wu, S., Ye, S. Y., Fischer, G., Taubenschuss, U., Jackman, C. M., O'Dwyer, E., et al. (2022a). Saturn anomalous myriametric radiation, a new type of saturn radio emission revealed by cassini. supplementary material: Catalogue of events. (Version 1.0) [Data set]. PADC. doi:10.25935/8MAY-4320
- Wu, S., Zarka, P., Lamy, L., Taubenschuss, U., Cecconi, B., Ye, S. Y., et al. (2022b). Catalogue of observations of the first harmonic of saturn kilometric radiation during cassini's grand finale (Version 1.0) [Data set]. PADC. doi:10.25935/T033-QS72
- Zarka, P., Treumann, R. A., Ryabov, B. P., and Ryabov, V. B. (2004). Radio and plasma waves at the outer planets. *Astrophys. Space Sci.* 33, 293–300. doi:10.1023/a:1012221527425
- Zhang, P. J., Wang, C. B., and Ye, L. (2018). A type III radio burst automatic analysis system and statistic results for a half solar cycle with Nançay Decameter Array data. *Astron. Astrophys.* 618, A165. doi:10.1051/0004-6361/201833260