

Extension ObsCore pour les données radio



F.Bonnarel

et le Radioastronomy Interest Group de
l'IVOA



ObsCore et découverte des données

- Idée : découvrir des données (datasets) en contraignant les valeurs de champs de métadonnées décrivant les données.
- ObsCore =
 - Provenance instrumentale (facility, instrument)
 - Identification (et appartenance à des collections)
 - Product type (image, cube, spectre, etc...)
 - Curation
 - Caractérisation des axes physiques (spatial, spectral, temporel , polarisation)
 - Paramètres d'accès aux données (url, format, datalink, cutout, etc..)
- **Le jeu standard est-il suffisant pour certaines catégories spécifiques de données ?**



Le jeu standard est-il suffisant pour certaines catégories spécifiques de données ?

- Oui sans doute pour des données réduites simples,
 - mais pas pour des données brutes,
 - ou des données à mode de réduction médiatisé par un modèle
- On l'a vu hier pour les données hautes énergie de type « event-list »
- Or on veut les exposer (avec accès à des softs de réduction)
- Certaines valeurs d'ObsCore sont difficiles à remplir pour ces données là



Le jeu standard est-il suffisant pour certaines catégories spécifiques de données ?

- Exemples en radio = l'amplitude spectrale d'un cube radio est telle que la résolution spatiale et le champ de vue ($\sim \lambda/D$) dépendent de la fréquence
 - $\rightarrow s_{\text{fov}}$ et $s_{\text{resolution}}$ sont des valeurs typiques insuffisantes pour sélectionner
 - $em_{\text{res_power}}$ (pouvoir de résolution : $\lambda/\Delta\lambda$) est peu usité car très variable
- En radio on aime bien les fréquences plutôt que les longueurs d'ondes
 - $\rightarrow em_{\text{min}}$ et em_{max} en longueur d'onde sont peu usités
- Observation à cibles multiples : même s_{ra} , s_{dec} sont difficiles à déterminer.
- Notion de « spectral windows » (fenêtres spectrales)



Une spécification DM pour une extension

- Statut : working draft
- Passer en « PR » avant l'interop ?
- Va dépendre de la réunion du 5 avril



IVOA Obscore Extension for Radio data Version 1.0

IVOA Working Draft 2024-03-15

Working Group

Data Model Working Group

This version

<https://www.ivoa.net/documents/ObsCoreExtensionForRadioData/20240315>

Latest version

<https://www.ivoa.net/documents/ObsCoreExtensionForRadioData>

Previous versions

Author(s)

François Bonnarel, Mireille Louys, Baptiste Cecconi, Vincenzo Galluzzi, Yan Grange, Mark Kettenis, Mark Lacy, Alan Loh, Mattia Mancini, Peter Teuben, Alessandra Zanichelli

Editor(s)

François Bonnarel, Mark Kettenis, Mireille Louys



ObsCore pour les données radio complexes

- Une Observation est faite de plusieurs datasets
 - Distinction obs_id et obs_publisher_id
 - Une ligne par obs_publisher_id, pas par observation
- Les caractérisations spatiales, spectrales et temporelles sont faites sur le dataset, pas sur l'observation
- Exemple : une observation interférométrique brute faite de
 - Une cible principale et un calibrateur
 - Deux fenêtres spectrales bien séparées et avec résolution spectrale différente
 - → 4 datasets avec le même obs_id (ce qui permettra de les regrouper au besoin)



Attributs ajoutés : axe spatial

- Minimum et maximum pour le champ de vue :
 - `s_fov_min` et `s_fov_max` (à chaque extrémité de la fenêtre spectrale)
- Minimum et maximum pour la résolution :
 - `s_resolution_min` et `s_resolution_max`
 - (à chaque extrémité de la fenêtre spectrale, dépend de la plus longue ligne de base)
- Échelle angulaire maximale pour les données interférométriques (filtre sur les basses fréquences spatiales)
 - `s_maximum_angular_scale` (dépend de la plus courte ligne de base)



Attributs ajoutés : axe spectral

- La résolution spectrale en fréquence est constante par fenêtre en radio, le pouvoir de résolution ne l'est pas
 - Introduction de `f_resolution`
- Proposition d'ajouter des bornes spectrales en fréquence
 - `f_min` et `f_max` aux cotés de `em_min` et `em_max`
 - C'est discuté, car certains ne veulent pas dupliquer l'info
 - User defined fonction pour exposer ou contraindre les fréquences
 - Désaccord des éditeurs
 - On a besoin de quelque chose de parlant pour le choix par l'utilisateur
 - Sera fait par une « view » de toutes façons
 - Ne marchera pas dans « DAP » (ex SIA)



Attributs ajoutés : détails instrumentaux

- Le mode de scan du ciel (voir page suivante pour le cas « antenne unique » - single dish)
- Le diamètre typique de l'antenne (des antennes)
 - Important pour le champ de vue
- En interférométrie
 - Le nombre d'antennes, la distance maximale et minimale entre les antennes
 - Important pour la qualité des données et la résolution
 - Discuté car information trop vague pour avoir une idée précise
 - Semble être une pratique de certains radio astronomes



Ce qui manque dans ObsCore : modes de scan du ciel

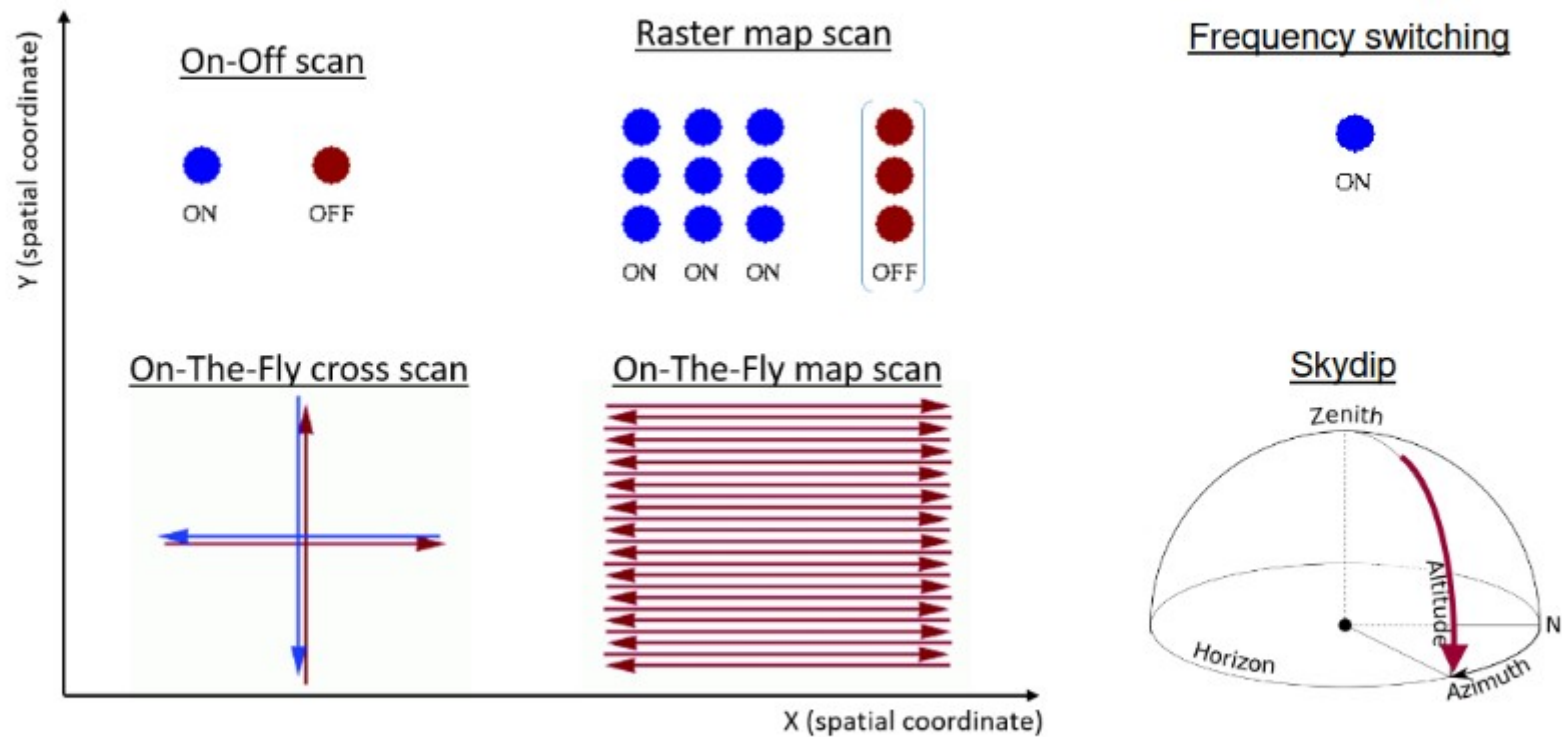


Figure 1: Single Dish Observation Sky scan modes



De nouveaux « dataproduct types »

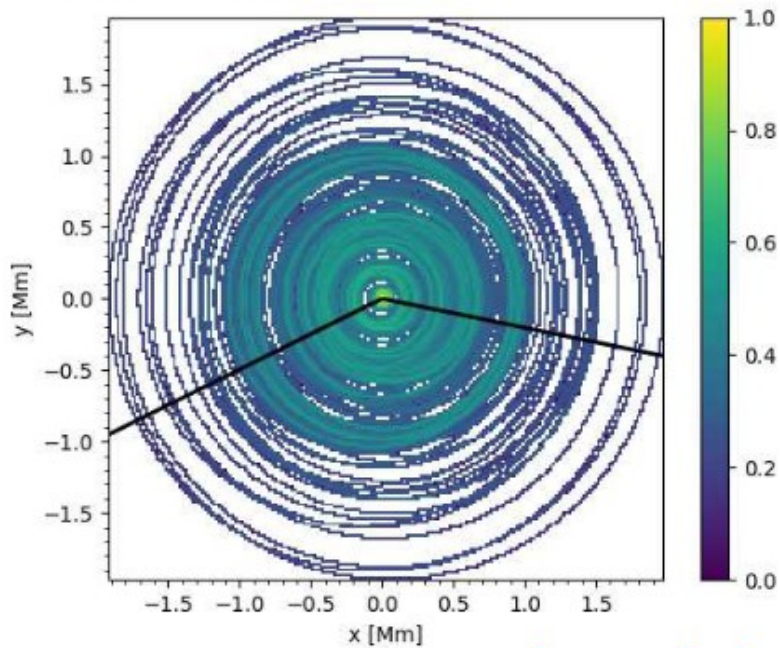
- Spectres dynamiques (déjà acté)
- Profil spatial (image 1 D) en cours d'adoption.
- Autres :
 - diagrammes position/vitesse ou position/fréquence
 - Velocity cubes
 - Cas des « scans maps » : balaye un champ du ciel mais les temps d'enregistrements sont différents en chaque point ? Comment les appeler ?



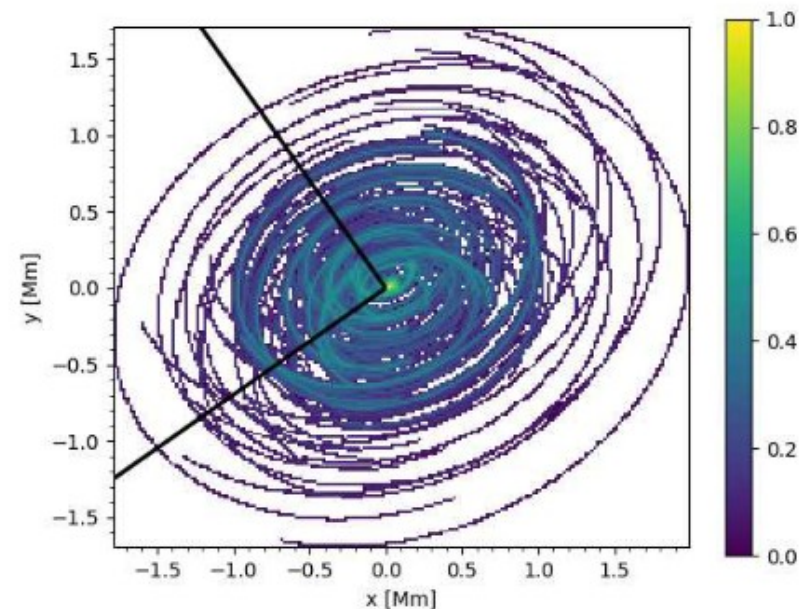
Ce qui manque dans ObsCore : caractérisation du plan uv

Interférométrie cartes de la couverture du plan uv :

In a well behaved case



In a more common case



https://git.astron.nl/virtualobservatory/lofar_uv_generator



Ce qui manque dans ObsCore : caractérisation du plan uv

- `uv_distance_min` , `uv_distance_max`
 - Importants pour la filtrage d'échelle et la résolution
- `uv_distribution_exc` : excentricité de la distribution
 - Régularité des données
- `uv_distribution_fill` : filling factor de la distribution
 - Échantillonnage des données



Accès à des cartes et diagrammes

- Via DataLink :
 - Distribution plan uv (uv coverage)
 - « Dirty beam »
 - Diagrammes amplitude/phase ou frequence/phase
- Pose la question du vocabulaire pour les décrire (dans content_qualifier) Dans semantics ce serait « auxiliary ».



Comment installer l'extension

- Deux façons possibles dans TAP:
 - Une table ObsCore de base, une table avec ObsCore+ extension
 - Une table ObsCore de base et une table extension.
L'utilisateur ou le client fait le join
- Conséquences sur le registry
- Idée de deux tables : une pour les détails configs/scans et une plus sur la caractérisation des données ???



Implementations et documents

- Implementation JIVE : Joint institute for VLBI ERIC
 - Ils ont déjà un service ObsCore pour leurs « visibilitées »
 - Sont en train d'ajouter l'extension
 - Implanté avec Dachs
- Prototypage CDS
 - Basé sur le service de découverte du SRC net.
 - Implanté avec dachs mais sera porté sur vollt
 - Conservé indépendamment de SKA
- Documents : <https://github.com/ivoa-std/ObsCoreExtensionForRadioData/>
and <https://ivoa.net/documents/ObsCoreExtensionForRadioData/20230512/index.html>
- → **Radio IG running meeting . Vendredi 5/04 à 14h CET**



Démo du prototype CDS (implanté avec Dachs) : les 16 datasets du service ObsCore test du SRCnetwork

The screenshot displays the Aladin v12.0 software interface, which is a BETA VERSION based on v12.033. The main window shows a star field with a command line at the top: `04:48:26.62 +30:29:11.4`. The interface includes a menu bar (File, Edit, Image, Catalog, Overlay, Coverage, Tool, View, Interop, Help) and a toolbar with various icons for image manipulation and data handling.

On the left side, there is a sidebar with "Available data" (35210) and "Collections" (35210). The "Collections" list includes: Image (621), Data base (4), Catalog (33146), Cube (24), Ancillary (82), Outreach (52), Others (1261), and Deprecated (20). Below this, the "Aladin Java measurements frame" is open, displaying a table of observations:

obs collection	obs id
MKT-MGCLS	Abell 194 I
MKT-MGCLS	Abell 194 Q
MKT-MGCM	Galactic Centre 1284MHz-StokesI
MKT-MGCM	Galactic Centre alpha
MKT-FORNAX-S...	MKT-FORNAX-SURV t06 1km NGC1436 image mos
ASK-RACS-DR1	RACS-DR1 0126+00A
ASK-WALLABY	Eridanus cutout NGC1436
LOF-LoTSS-DR2	P39Hetdex19
Ape-DR1	200426041 AP B021
Ape-DR1	200426041 AP B021
VLA-VLASS	T10t02.i005000-023000.06.2048
VLA-VLASS	T10t02.i005000-023000.06.2048
SDC01	SDC01 SKAMid B2 1000h
SDC02	SDC02 SKAMid sky ldev v2
SDC03	SDC03 ZW3.msn

The "Server selector" dialog box is open, showing the "Others" tab. It displays the server "vo-proto-debian.cds.unistra.fr" in "Mode: Generic". The query editor shows the following query: `SELECT TOP 9999 * FROM rucio.obscore`. The "Table" is set to "rucio.obscore" and the "Constraints" are "Add new". The "Max rows" is set to "9999". The "Target" is "6024000 -02 15 16.848000" and the "Radius" is "180°". The "Refresh query" button is highlighted.

At the bottom of the interface, there is a status bar with the text "Fichier Édition Affichage Historique Marque-pages Outils Aide".

Démo du prototype CDS (implanté avec Dachs) :

frequency between 1 and 2 Ghz (upper left - 3 results)

freq. between 1 and 2 Ghz and spectral resolution better than 100 Mhz (lower right - 2 results)

The screenshot shows the Aladin v12.0 interface. A 'Server selector' dialog box is open, displaying a query construction interface. The query is: `SELECT obs_publisher_id, target_name, s_ra, s_dec, s_fov, s_resolution, em_min, em_max, s_fov_min, s_fov_max, s_resolution_min, s_resolution_max, f_resolution, f_min, f_max, instrument_ant_max_dist, instrument_ant_diameter FROM rucio.obscore where f_min > 1e9 and f_max < 2e9`. Below the dialog, a table of search results is visible:

s resolution	em min	em max	s fov min...	s fov max...	s resolution min	s resolution max	f resolution	f min	f max
0.12	2.4	0.212056998	0.212333965	0.119900003075	0.120099999010	2.39800001907	6494	1412888608	1414713984
1.1	18	0.22438	0.2321	1.080000042915	1.120000004768	17.700000076293	3567	192549304	1337017800
1	0.6	0.106	0.252	0.850000023841	1.149999976151	0.509999990463	411370272	1190467290	1612903168

SELECT obs_publisher_id,
target_name, s_ra, s_dec, s_fov,
s_resolution, em_min, em_max,
s_fov_min, s_fov_max, s_resolution_min,
s_resolution_max, f_resolution, f_min,
f_max, instrument_ant_max_dist,
instrument_ant_diameter
FROM rucio.obscore
where f_min > 1e9 and f_max < 2e9

SELECT obs_publisher_id, target_name,
s_ra, s_dec, s_fov, s_resolution, em_min, em_max,
s_fov_min, s_fov_max, s_resolution_min,
s_resolution_max, f_resolution, f_min, f_max,
instrument_ant_max_dist, instrument_ant_diameter
FROM rucio.obscore
where f_min > 1e9 and f_max < 2e9
and f_resolution < 1e8

The screenshot shows the Aladin v12.0 interface with a 'Server selector' dialog box. The query is: `SELECT obs_publisher_id, target_name, s_ra, s_dec, s_fov, s_resolution, em_min, em_max, s_fov_min, s_fov_max, s_resolution_min, s_resolution_max, f_resolution, f_min, f_max, instrument_ant_max_dist, instrument_ant_diameter FROM rucio.obscore where f_min > 1e9 and f_max < 2e9 and f_resolution < 1e8`.

The screenshot shows the Aladin v12.0 interface with a table of search results:

obs_publisher_id	target_name	s ra	s dec	s fov	s resolution	em min	em max	s fov min...	s fov max...	s resolution min	s resolution max	f resolution	f min	f max
no/rest.skae/~? NGC1436		59.9045	-39.853028	0.12	2.4	0.212056998	0.212333965	0.119900003075	0.120099999010	2.39800001907	2.39800001907	6494	1412888608	1414713984
no/rest.skae/~? 11123+5550		169.447269	56.014556	1.1	18	0.22438	0.2321	1.080000042915	1.120000004768	17.700000076293	17.700000076293	3567	192549304	1337017800

The screenshot shows the Aladin v12.0 interface with a table of search results:

obs_publisher_id	target_name	s ra	s dec	s fov	s resolution	em min	em max	s fov min...	s fov max...	s resolution min	s resolution max	f resolution	f min	f max
no/rest.skae/~? NGC1436		59.9045	-39.853028	0.12	2.4	0.212056998	0.212333965	0.119900003075	0.120099999010	2.39800001907	2.39800001907	6494	1412888608	1414713984
no/rest.skae/~? 11123+5550		169.447269	56.014556	1.1	18	0.22438	0.2321	1.080000042915	1.120000004768	17.700000076293	17.700000076293	3567	192549304	1337017800

Démo du prototype CDS (implanté avec Dachs) :

s_resolution better than 0.6 arcsec (upper left - no result)

s_resolution_min better than 0.6 arcsec (lower right - 1 result)

The screenshot shows the ALADIN interface with a query window open. The query is: `s_resolution_min, s_resolution_max, f_resolution, f_min, f_max, instrument_ant_max_dist, instrument_ant_diameter FROM rucio.obscore WHERE s_resolution < 0.6`. The interface shows a table with columns for 'Table', 'Select', 'Constraints', and 'Max rows'. The 'Table' is set to 'rucio.obscore'. The 'Select' field contains the query. The 'Constraints' field is empty. The 'Max rows' field is set to 'All'. The 'Refresh query' button is visible. The interface also shows a 'Server selector' window with 'vo-proto-debian.cds.unistra.fr' selected. The main view shows a dark sky image with a red crosshair.

```
SELECT obs_publisher_did,  
target_name, s_ra, s_dec, s_fov,  
s_resolution, em_min, em_max,  
s_fov_min, s_fov_max, s_resolution_min,  
s_resolution_max, f_resolution, f_min,  
f_max, instrument_ant_max_dist,  
instrument_ant_diameter  
FROM rucio.obscore  
where s_resolution < 0.6
```

```
SELECT obs_publisher_did, target_name, s_ra,  
s_dec, s_fov, s_resolution, em_min, em_max,  
s_fov_min, s_fov_max, s_resolution_min,  
s_resolution_max, f_resolution, f_min, f_max,  
instrument_ant_max_dist, instrument_ant_diameter  
FROM rucio.obscore  
where s_resolution_min < 0.6
```

→ At least some part of the dataset has a resolution better than 0.6 arcsec

The screenshot shows the ALADIN interface with a query window open. The query is: `s_resolution_min, s_resolution_max, f_resolution, f_min, f_max, instrument_ant_max_dist, instrument_ant_diameter FROM rucio.obscore WHERE s_resolution_min < 0.6`. The interface shows a table with columns for 'Table', 'Select', 'Constraints', and 'Max rows'. The 'Table' is set to 'rucio.obscore'. The 'Select' field contains the query. The 'Constraints' field is empty. The 'Max rows' field is set to 'All'. The 'Refresh query' button is visible. The interface also shows a 'Server selector' window with 'vo-proto-debian.cds.unistra.fr' selected. The main view shows a dark sky image with a red crosshair. A table of results is visible at the bottom of the interface.

min	em_max	s_fov_min	s_fov_max	s_resolution_min	s_resolution_max	f_resolution	f_min	f_max	instrument_ant_max_dist	instrument_ant_diameter
0.186	0.252	0.552000923841	1.14999997615	0.3099999804632568	0.689999976158142	4.11370272	1169467200	1017393188	75266	12.53699994

Démo du prototype CDS (implanté avec Dachs) :

`s_fov` larger than 7.5 deg (upper left – two results)

`s_fov_min` larger than 7.5 deg (lower right – 1 result)

```
SELECT obs_publisher_did,
target_name, s_ra, s_dec, s_fov,
s_resolution, em_min, em_max,
s_fov_min, s_fov_max, s_resolution_min,
s_resolution_max, f_resolution, f_min,
f_max, instrument_ant_max_dist,
instrument_ant_diameter
FROM rucio.obscure
where s_fov > 7.5 deg
```

FROM rucio.obscure
where s_fov > 7.5 deg

```
SELECT obs_publisher_did, target_name, s_ra,
s_dec, s_fov, s_resolution, em_min, em_max,
s_fov_min, s_fov_max, s_resolution_min,
s_resolution_max, f_resolution, f_min, f_max,
instrument_ant_max_dist, instrument_ant_diameter
FROM rucio.obscure
where s_fov_min > 7.5
```

→ all channels of the dataset have a field of view larger than 7.5