

# 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  $em_{\text{max}}$  en longueur d'onde sont peu usités
- Observation à cibles multiples : même  $s_{\text{ra}}$ ,  $s_{\text{dec}}$  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

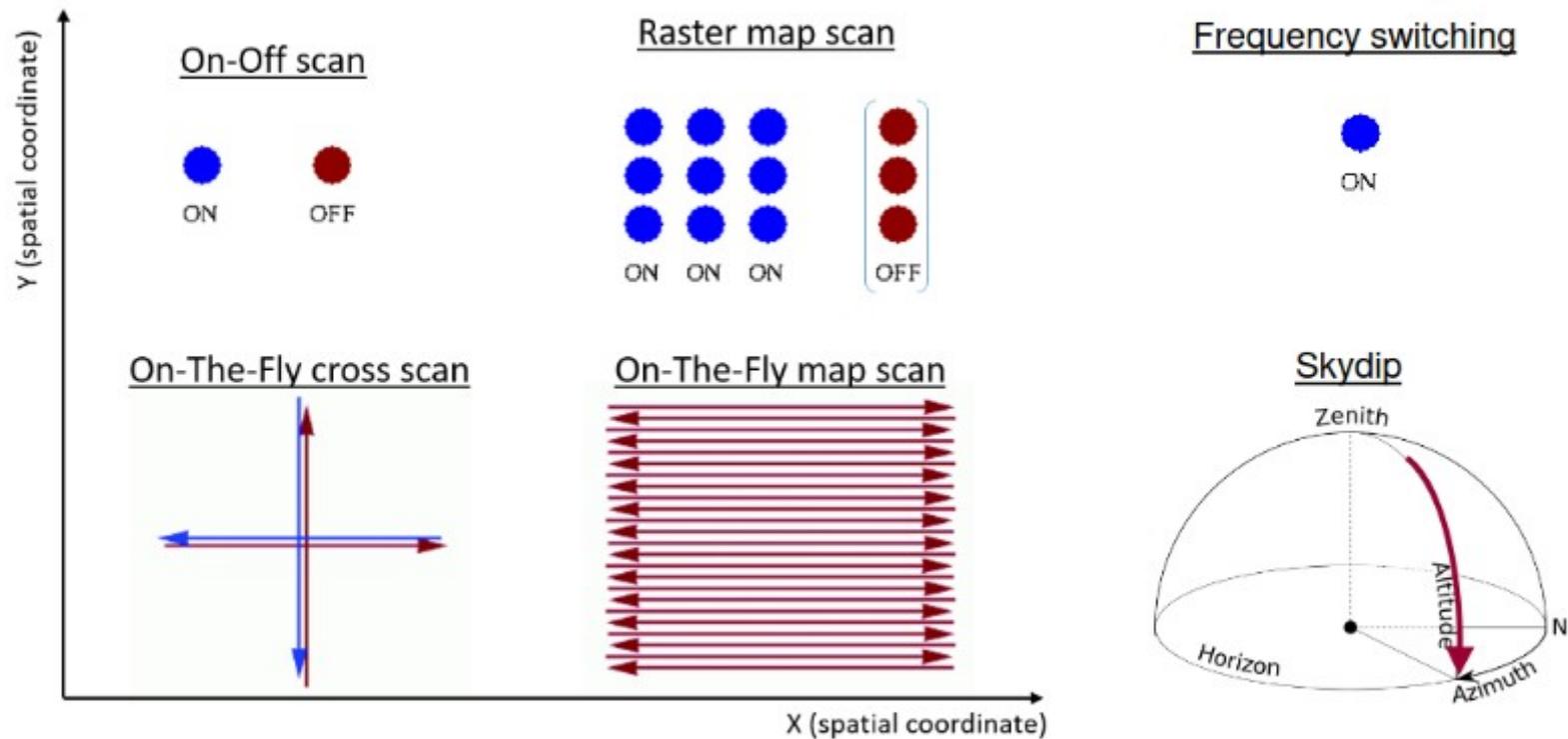


Figure 1: Single Dish Observation Sky scan modes



# De nouveaux « dataproduct types »

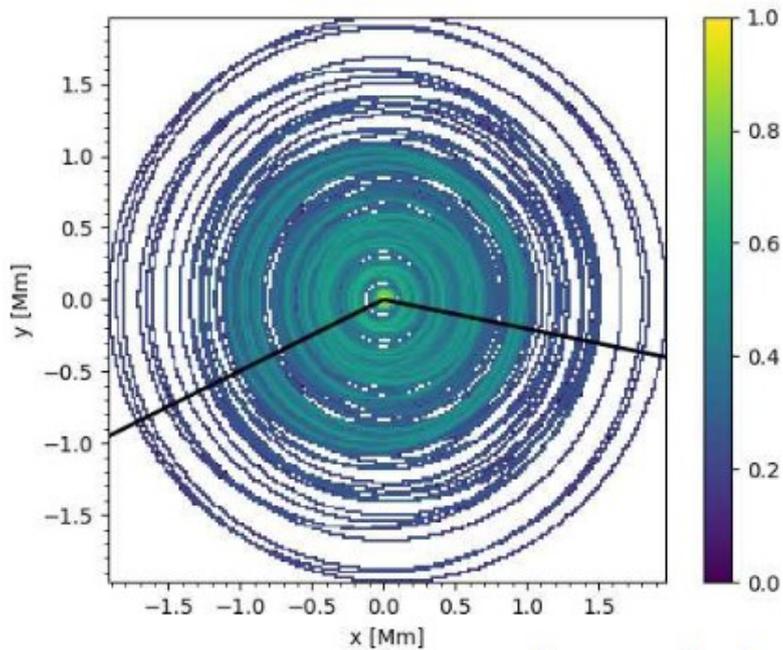
- Spectres dynamiques (déjà acté)
- Profil spatial (image 1 D) en cours d'adoption.
- Autres :
  - diagrammes position/vitesse ou position/fréquence
  - Velocity cubes
  - Cas des « scans maps » : balaye un champ du ciel mais les temps d'enregistrements sont différents en chaque point ? Comment les appeler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



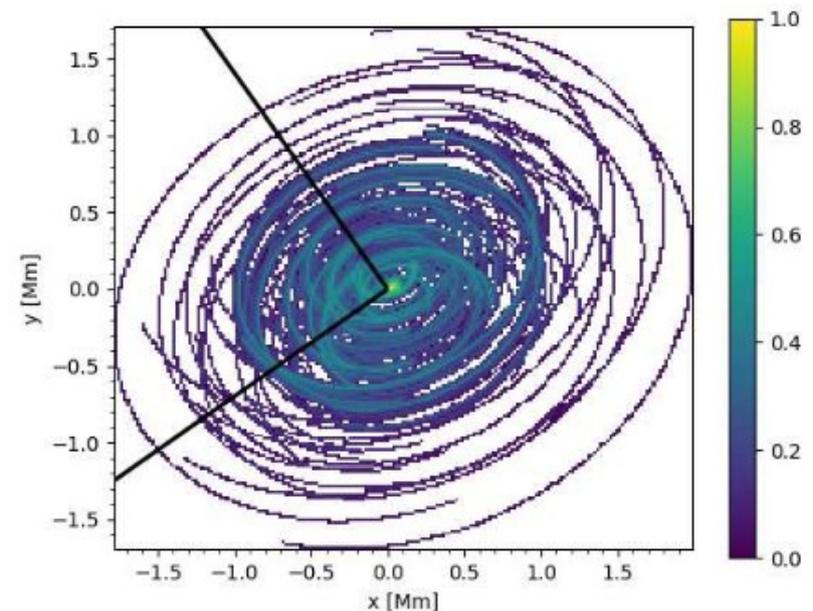
# Ce qui manque dans ObsCore : caractérisation du plan uv

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](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 command line is followed by a list of datasets: `DSS PanSTARRS SDSS 2MASS GALEX Gaia Simbad NED`. The main window is titled "DSS2 color".

In the bottom-left corner, there is a table titled "Aladin Java measurements frame" with a search bar. The table contains the following data:

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

In the bottom-right corner, there is a "Server selector" dialog box. It shows the server "vo-proto-debian.cds.unistra.fr" and the table "rucio.obscore". The query editor contains the following SQL query:

```
SELECT TOP 9999 * FROM rucio.obscore
```

The dialog box also includes a "Target" field with the value "6024000 -02 15 16.848000" and a "Radius" field with the value "180°".

# 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 for 'vo-proto-debian.cds.unistra.fr'. The query is: `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 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	0,120089999010	2,39800001907	6484	1412888608	1414713984
1,1	18	0,22438	0,2321	1,080000042915	1,120000004768	17,70000076281	3567	192549304	1337017800
1	0,6	0,106	0,252	0,850000023841	1,148999976158	0,508999990483	411370272	1190467290	1612903168

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 f\_min > 1e9 and f\_max < 2e9

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 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_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 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_d	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
no/rest.skae/~? NGC1436		59.9045	-39.853028	0,12	2,4	0,212056998	0,212333965	0,119900003075	0,120089999010	2,39800001907	2,401999950408
no/rest.skae/~? 1123+5550		168.447269	56.014556	1,1	18	0,22438	0,2321	1,080000042915	1,120000004768	17,70000076281	18,29999923706

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

obs_publisher_d	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
no/rest.skae/~? NGC1436		59.9045	-39.853028	0,12	2,4	0,212056998	0,212333965	0,119900003075	0,120089999010	2,39800001907	2,401999950408
no/rest.skae/~? 1123+5550		168.447269	56.014556	1,1	18	0,22438	0,2321	1,080000042915	1,120000004768	17,70000076281	18,29999923706

# 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)

Available data → 35212

Command: 2024-03-08 22:45:08.6

Server selector: vo-proto-debian.cds.unistra.fr

Table: rucio.obscure

Select: All

Constraints: Add new

Max rows: 100

Target: 9990100+16 09 39.917900

Radius: 180°

Refresh query

Check... SYNC Async jobs>>

Reset Clear SUBMIT Close

s\_resolution\_min, s\_resolution\_max, f\_resolution, f\_min, f\_max, instrument\_ant\_max\_dist, instrument\_ant\_diameter FROM rucio.obscure WHERE s\_resolution < 0.6

```
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.obscure
where s_resolution < 0.6
```

```
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.obscure
where s_resolution_min < 0.6
```

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

Server selector: vo-proto-debian.cds.unistra.fr

Table: rucio.obscure

Select: All

Constraints: Add new

Max rows: 100

Target: 9990100+16 09 39.917900

Radius: 180°

Refresh query

Check... SYNC Async jobs>>

Reset Clear SUBMIT Close

s\_resolution\_min, s\_resolution\_max, f\_resolution, f\_min, f\_max, instrument\_ant\_max\_dist, instrument\_ant\_diameter FROM rucio.obscure WHERE s\_resolution\_min < 0.6

id	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
0.188	0.252	0.552000923841	1.14999997915	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

Aladin v12.0 \*\*\* BETA VERSION (based on v12.033) \*\*\*

Command: `s_fov > 7.5`

Server selector: `vo-proto-debian.cds.unistra.fr`

Table: rucio.obscure

Select: `s_fov > 7.5`

Constraints: Add new

Max rows: 100

Target: 1872200 -19 18 33.425300

Radius: 180°

Refresh query Check.. SYNC Async jobs>>

Reset Clear SUBMIT Close

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
ivo/test.skao/~71	0126+00	21.722065416671	0.003423722222	9.5	25	0.338	0.338	9.5	9.5	25	25
ivo/test.skao/~71	0.0-30.0	0	-30	9.1	16	1.53	2.83	6.3800011444...	11.8100004196...	11.22989954223...	20

Aladin v12.0 \*\*\* BETA VERSION (based on v12.033) \*\*\*

Command: `s_fov_min > 7.5`

Server selector: `vo-proto-debian.cds.unistra.fr`

Table: rucio.obscure

Select: `s_fov_min > 7.5`

Constraints: Add new

Max rows: 100

Target: 2957000 +00 00 12.325400

Radius: 9.899°

Refresh query Check.. SYNC Async jobs>>

Reset Clear SUBMIT Close

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
ivo/test.skao/~71	0126+00	21.722065416671	0.003423722222	9.5	25	0.338	0.338	9.5	9.5	25	25

```
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