Australian Government

ANSTO

What is a dataset?

James Hester

Australian Centre for Neutron Scattering

Science. Ingenuity. Sustainability.

FAIR

- (F)indable: Keywords, semantic web, registries, DOIs
- (A)ccessible: DOIs, protocols
- (I)nteroperable: Standards for data description
- (R)eusable: Standards

The "data library"

You've found the book: now what?

What is a dataset?

- All the data required to support a (published) result?
- All the data collected during an experimental run?
- All the data necessary to assess a theoretical calculation?
- …

A dataset is the complete data required for a particular purpose

How do we describe datasets to software?

- Do not necessarily coincide with the contents of a DOI
  - A data collection split over several DOIs
  - Several datasets in a DOI data dump
- File formats vary and can be mixed together

A motivating example: CheckCIF for raw data

Check raw data files for

- consistent metadata
- sufficient metadata
- agreement with results

But . . .
Raw data diffraction files come in over 200 different styles (V. Minor)

Solution

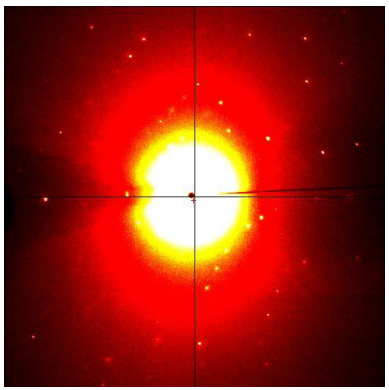
A collection of data files = a collection of tables

(see Hester, J.R. "A robust, format-agnostic scientific data transfer framework", (2016) *Data Science Journal* 15 12:1-17)

A dataset specification = a set of column headings

First: map file contents to tabular form

```
{
HEADER_BYTES=  512;
DIM=2;
BYTE_ORDER=little_endian;
TYPE=unsigned_short;
SIZE1=2048;
SIZE2=2048;
PIXEL_SIZE=0.1024;
BIN=2x2;
ADC=fast;
DETECTOR_SN=457;
TIME=2.000000;
DISTANCE=68.042500;
PHI=4.000000;
OSC_START=4.000000;
OSC_RANGE=2.000000;
WAVELENGTH=0.710894;
BEAM_CENTER_X=105.134680;
BEAM_CENTER_Y=104.653206;
ACC_TIME=3156;
CREV=1;
BIN_TYPE=HW;
```



ADSC (SMV) format

- One file per oscillation step
- Header + data

ADSC files as a collection of tables

Frame_id	Time	Detector_SN	Wavelength
kt_001	2.0	457	0.710894
kt_002	2.0	457	0.710894
kt_003	2.0	457	0.710894

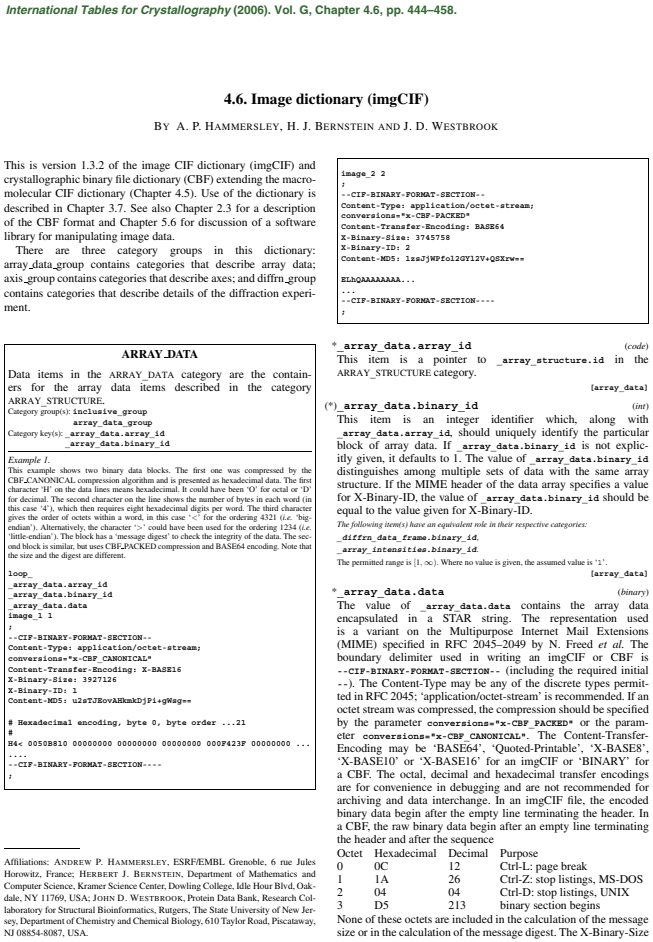
Frame_id	Axis	Displacement	Angle
kt_001	phi	.	0.0
kt_001	dist	68.042	.
kt_002	phi	.	2.0
kt_002	dist	68.042	.
kt_003	phi	.	4.0
kt_003	dist	68.042	.

Grouping data files

Frame_id	Time	Detector_SN	Wavelength	Scan_id
kt_001	2.0	457	0.710894	1
kt_002	2.0	457	0.710894	1
kt_003	2.0	457	0.710894	1

Frame_id	Axis	Displacement	Angle
kt_001	phi	.	0.0
kt_001	dist	68.042	.
kt_002	phi	.	2.0
kt_002	dist	68.042	.
kt_003	phi	.	4.0
kt_003	dist	68.042	.

imgCIF: A "master" image ontology



International Tables for Crystallography (2006). Vol. G, Chapter 4.6, pp. 444–458.

4.6. Image dictionary (imgCIF)

BY A. P. HAMMERSLEY, H. J. BERNSTEIN AND J. D. WESTBROOK

This is version 1.3.2 of the image CIF dictionary (imgCIF) and crystallographic binary file dictionary (CBF) extending the macromolecular CIF dictionary (Chapter 4.5). Use of the dictionary is described in Chapter 3.7. See also Chapter 2.3 for a description of the CBF format and Chapter 5.6 for discussion of a software library for manipulating image data.

There are three category groups in this dictionary: array_data_group contains categories that describe array data; axis_group contains categories that describe axes; and diffrn_group contains categories that describe details of the diffraction experiment.

ARRAY_DATA

Data items in the ARRAY_DATA category are the containers for the array data items described in the category ARRAY_STRUCTURE.

Category group(s): `inclusive_group`
`array_data_group`
Category key(s): `_array_data.array_id`
`_array_data.binary_id`

Example 1.
This example shows two binary data blocks. The first one was compressed by the CBF_CANONICAL compression algorithm and is presented as hexadecimal data. The first character 'H' on the data lines means hexadecimal. It could have been 'O' for octal or 'D' for decimal. The second character on the line shows the number of bytes in each word (in this case '4'), which then requires eight hexadecimal digits per word. The third character gives the order of octets within a word, in this case '<' for the ordering 4321 (*i.e.* 'big-endian'). Alternatively, the character '>' could have been used for the ordering 1234 (*i.e.* 'little-endian'). The block has a 'message digest' to check the integrity of the data. The second block is similar, but uses CBF_PACKED compression and BASE64 encoding. Note that the size and the digest are different.

```
loop_
_array_data.array_id
_array_data.binary_id
_array_data.data
image_1 1
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
conversions="x-CBF_CANONICAL"
Content-Transfer-Encoding: X-BASE16
X-Binary-Size: 3927126
X-Binary-ID: 1
Content-MD5: u2sTJEovAHkmkDjPi+gWag==

# Hexadecimal encoding, byte 0, byte order ...21
#
H4< 0050B810 00000000 00000000 00000000 000F423F 00000000 ...
....
--CIF-BINARY-FORMAT-SECTION----
;
```

```
image_2 2
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
conversions="x-CBF-PACKED"
Content-Transfer-Encoding: BASE64
X-Binary-Size: 3745758
X-Binary-ID: 2
Content-MD5: 1zsJjWPfol2GY12V+QSXrw==

ELbQAAAAAAAA...
...
--CIF-BINARY-FORMAT-SECTION----
;
```

* `_array_data.array_id` (*code*)
This item is a pointer to `_array_structure.id` in the ARRAY_STRUCTURE category.

`[array_data]`

(*) `_array_data.binary_id` (*int*)
This item is an integer identifier which, along with `_array_data.array_id`, should uniquely identify the particular block of array data. If `_array_data.binary_id` is not explicitly given, it defaults to 1. The value of `_array_data.binary_id` distinguishes among multiple sets of data with the same array structure. If the MIME header of the data array specifies a value for X-Binary-ID, the value of `_array_data.binary_id` should be equal to the value given for X-Binary-ID.

The following item(s) have an equivalent role in their respective categories:
`_diffrn_data_frame.binary_id`,
`_array_intensities.binary_id`.

The permitted range is [1, ∞). Where no value is given, the assumed value is '1'.

`[array_data]`

* `_array_data.data` (*binary*)
The value of `_array_data.data` contains the array data encapsulated in a STAR string. The representation used is a variant on the Multipurpose Internet Mail Extensions (MIME) specified in RFC 2045–2049 by N. Freed *et al.* The boundary delimiter used in writing an imgCIF or CBF is `--CIF-BINARY-FORMAT-SECTION--` (including the required initial `--`). The Content-Type may be any of the discrete types permitted in RFC 2045; 'application/octet-stream' is recommended. If an octet stream was compressed, the compression should be specified by the parameter `conversions="x-CBF_PACKED"` or the parameter `conversions="x-CBF_CANONICAL"`. The Content-Transfer-Encoding may be 'BASE64', 'Quoted-Printable', 'X-BASE8', 'X-BASE10' or 'X-BASE16' for an imgCIF or 'BINARY' for a CBF. The octal, decimal and hexadecimal transfer encodings are for convenience in debugging and are not recommended for archiving and data interchange. In an imgCIF file, the encoded binary data begin after the empty line terminating the header. In a CBF, the raw binary data begin after an empty line terminating the header and after the sequence

Octet	Hexadecimal	Decimal	Purpose
0	0C	12	Ctrl-L: page break
1	1A	26	Ctrl-Z: stop listings, MS-DOS
2	04	04	Ctrl-D: stop listings, UNIX
3	D5	213	binary section begins

None of these octets are included in the calculation of the message size or in the calculation of the message digest. The X-Binary-Size

Affiliations: ANDREW P. HAMMERSLEY, ESRF/EMBL Grenoble, 6 rue Jules Horowitz, France; HERBERT J. BERNSTEIN, Department of Mathematics and Computer Science, Kramer Science Center, Dowling College, Idle Hour Blvd, Oakdale, NY 11769, USA; JOHN D. WESTBROOK, Protein Data Bank, Research Collaboratory for Structural Bioinformatics, Rutgers, The State University of New Jersey, Department of Chemistry and Chemical Biology, 610 Taylor Road, Piscataway, NJ 08854-8087, USA.

Ontologies

A collection of definitions for data names

imgCIF describes a collection of tables


```
# category DIFFRN_DATA_FRAME
loop_
_diffrn_data_frame.id
_diffrn_data_frame.detector_element_id
_diffrn_data_frame.array_id
_diffrn_data_frame.binary_id
  FRAME1 ELEMENT1 ARRAY1 1

# category DIFFRN_MEASUREMENT
loop_
_diffrn_measurement.diffrn_id
_diffrn_measurement.id
_diffrn_measurement.number_of_axes
_diffrn_measurement.method
  P6MB GONIOMETER 3 rotation

# category DIFFRN_MEASUREMENT_AXIS
loop_
_diffrn_measurement_axis.measurement_id
_diffrn_measurement_axis.axis_id
  GONIOMETER GONIOMETER_PHI
  GONIOMETER GONIOMETER_KAPPA
  GONIOMETER GONIOMETER_OMEGA

# category DIFFRN_SCAN
loop_
_diffrn_scan.id
_diffrn_scan.frame_id_start
_diffrn_scan.frame_id_end
_diffrn_scan.frames
  SCAN1 FRAME1 FRAME1 1

# category DIFFRN_SCAN_AXIS
loop_
_diffrn_scan_axis.scan_id
_diffrn_scan_axis.axis_id
_diffrn_scan_axis.angle_start
_diffrn_scan_axis.angle_range
_diffrn_scan_axis.angle_increment
_diffrn_scan_axis.displacement_start
_diffrn_scan_axis.displacement_range
_diffrn_scan_axis.displacement_increment
  SCAN1 GONIOMETER_OMEGA 12.0 1.0 1.0 0.0 0.0 0.0
  SCAN1 GONIOMETER_KAPPA 23.3 0.0 0.0 0.0 0.0 0.0
  SCAN1 GONIOMETER_PHI -165.8 0.0 0.0 0.0 0.0 0.0
  SCAN1 DETECTOR_Z 0.0 0.0 0.0 -240.0 0.0 0.0
  SCAN1 DETECTOR_Y 0.0 0.0 0.0 0.6 0.0 0.0
  SCAN1 DETECTOR_X 0.0 0.0 0.0 -0.5 0.0 0.0
  SCAN1 DETECTOR_PITCH 0.0 0.0 0.0 0.0 0.0 0.0

# category DIFFRN_SCAN_FRAME
loop_
_diffrn_scan_frame.frame_id
_diffrn_scan_frame.frame_number
_diffrn_scan_frame.integration_time
_diffrn_scan_frame.scan_id
_diffrn_scan_frame.date
  FRAME1 1 20.0 SCAN1 1997-12-04T10:23:48

# category DIFFRN_SCAN_FRAME_AXIS
loop_
_diffrn_scan_frame_axis.frame_id
_diffrn_scan_frame_axis.axis_id
_diffrn_scan_frame_axis.angle
_diffrn_scan_frame_axis.displacement
  FRAME1 GONIOMETER_OMEGA 12.0 0.0
  FRAME1 GONIOMETER_KAPPA 23.3 0.0
  FRAME1 GONIOMETER_PHI -165.8 0.0
  FRAME1 DETECTOR_Z 0.0 -240.0
  FRAME1 DETECTOR_Y 0.0 0.6
  FRAME1 DETECTOR_X 0.0 -0.5
  FRAME1 DETECTOR_PITCH 0.0 0.0

# category AXIS
loop_
_axis.id _axis.type _axis.equipment _axis.depends_on
_axis.vector[1] _axis.vector[2] _axis.vector[3]
_axis.offset[1] _axis.offset[2] _axis.offset[3]
GONIOMETER_OMEGA rotation goniometer . 1 0 0 . . .
GONIOMETER_KAPPA rotation goniometer GONIOMETER_OMEGA
     0.64279 0 0.76604 . . .
```

```
GONIOMETER_PHI rotation goniometer GONIOMETER_KAPPA
     1 0 0 . . .
SOURCE          general source . 0 0 1 . . .
GRAVITY         general gravity . 0 -1 0 . . .
DETECTOR_Z      translation detector . 0 0 1 0 0 0
DETECTOR_Y      translation detector DETECTOR_Z 0 1 0 0 0 0
DETECTOR_X      translation detector DETECTOR_Y 1 0 0 0 0 0
DETECTOR_PITCH rotation    detector DETECTOR_X 0 1 0 0 0 0
ELEMENT_ROT   translation detector DETECTOR_PITCH 0 0 1 0 0 0
ELEMENT_RAD    translation detector ELEMENT_ROT 0 1 0 0 0 0

# category ARRAY_STRUCTURE_LIST
loop_
_array_structure_list.array_id
_array_structure_list.index
_array_structure_list.dimension
_array_structure_list.precedence
_array_structure_list.direction
_array_structure_list.axis_set_id
  ARRAY1 1 8309900 1 increasing ELEMENT_SPIRAL

# category ARRAY_STRUCTURE_LIST_AXIS
loop_
_array_structure_list_axis.axis_set_id
_array_structure_list_axis.axis_id
_array_structure_list_axis.angle
_array_structure_list_axis.displacement
_array_structure_list_axis.angular_pitch
_array_structure_list_axis.radial_pitch
  ELEMENT_SPIRAL ELEMENT_ROT 0    .  0.075   .
  ELEMENT_SPIRAL ELEMENT_RAD . 172.5  .    -0.150

# category ARRAY_ELEMENT_SIZE
# The actual pixels are 0.075 by 0.150 mm.
# We give the coarser dimension here.
loop_
_array_element_size.array_id
_array_element_size.index
_array_element_size.size
  ARRAY1 1 150e-6

# category ARRAY_INTENSITIES
loop_
_array_intensities.array_id
_array_intensities.binary_id
_array_intensities.linearity
_array_intensities.gain
_array_intensities.gain_esd
_array_intensities.overload
_array_intensities.undefined_value
  ARRAY1 1 linear 1.15 0.2 240000 0

# category ARRAY_STRUCTURE
loop_
_array_structure.id
_array_structure.encoding_type
_array_structure.compression_type
_array_structure.byte_order
  ARRAY1 "signed 32-bit integer" packed little_endian

# category ARRAY_DATA
loop_
_array_data.array_id
_array_data.binary_id
_array_data.data
 ARRAY1 1
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
conversions="x-CBF_PACKED"
Content-Transfer-Encoding: BASE64
X-Binary-Size: 3801324
X-Binary-ID: 1
X-Binary-Element-Type: "signed 32-bit integer"
Content-MD5: 071ZFvF+aOcW85IN7us18A==

AABRAAAAAAAAAAAAAAAAAAAA ...AAZBQSrlsKNBOeOe9HITdMdDUnbq7bg
 ...
8REo6TtBrxJlvKqAvx9YDMD6...r/tgssjMIJMXATDsZobL90AEXc4KigE

--CIF-BINARY-FORMAT-SECTION----
;
```

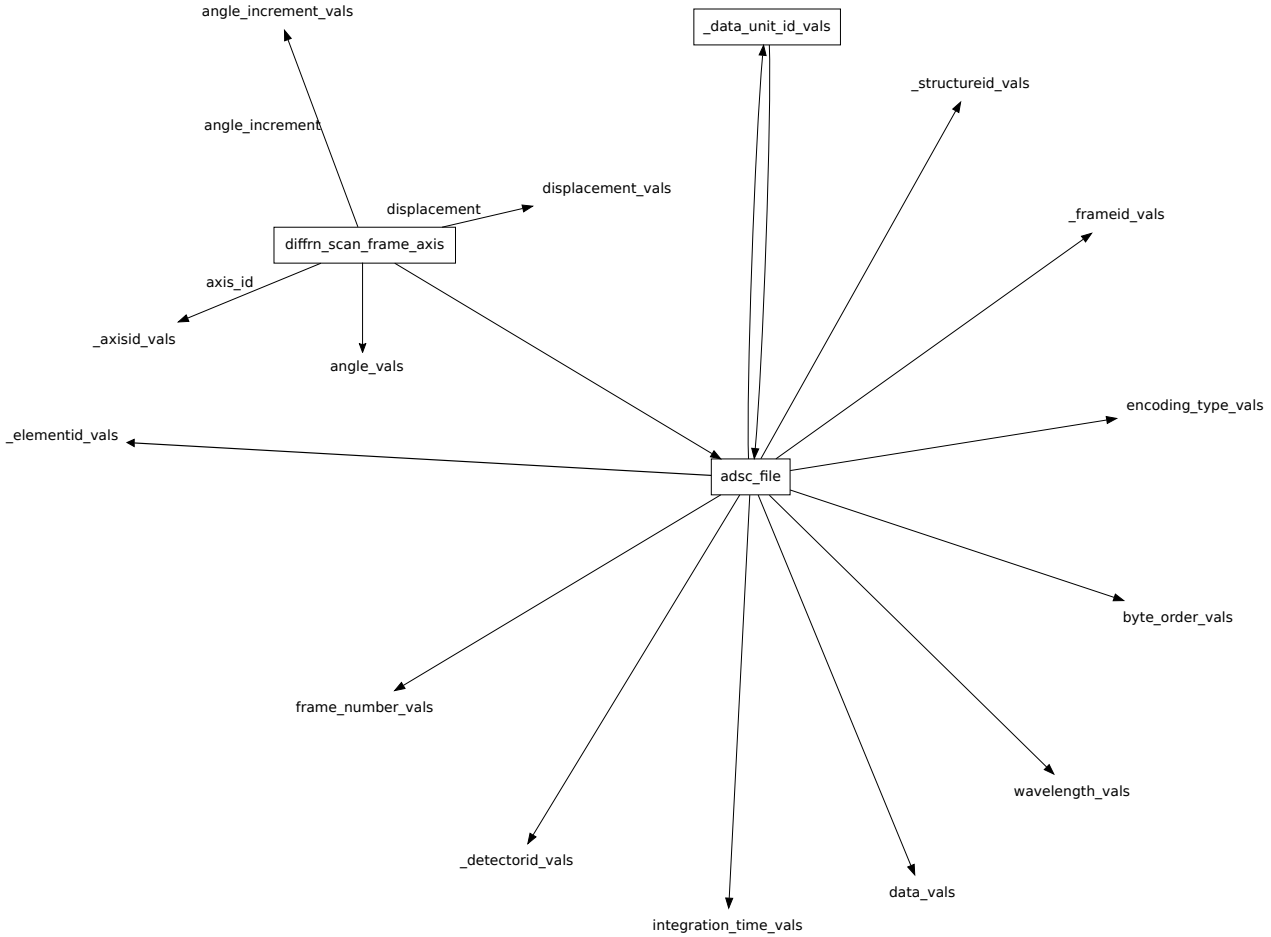
- imgCIF data names = column headings
- *are they compatible with our non-imgCIF data files?*

Category theory

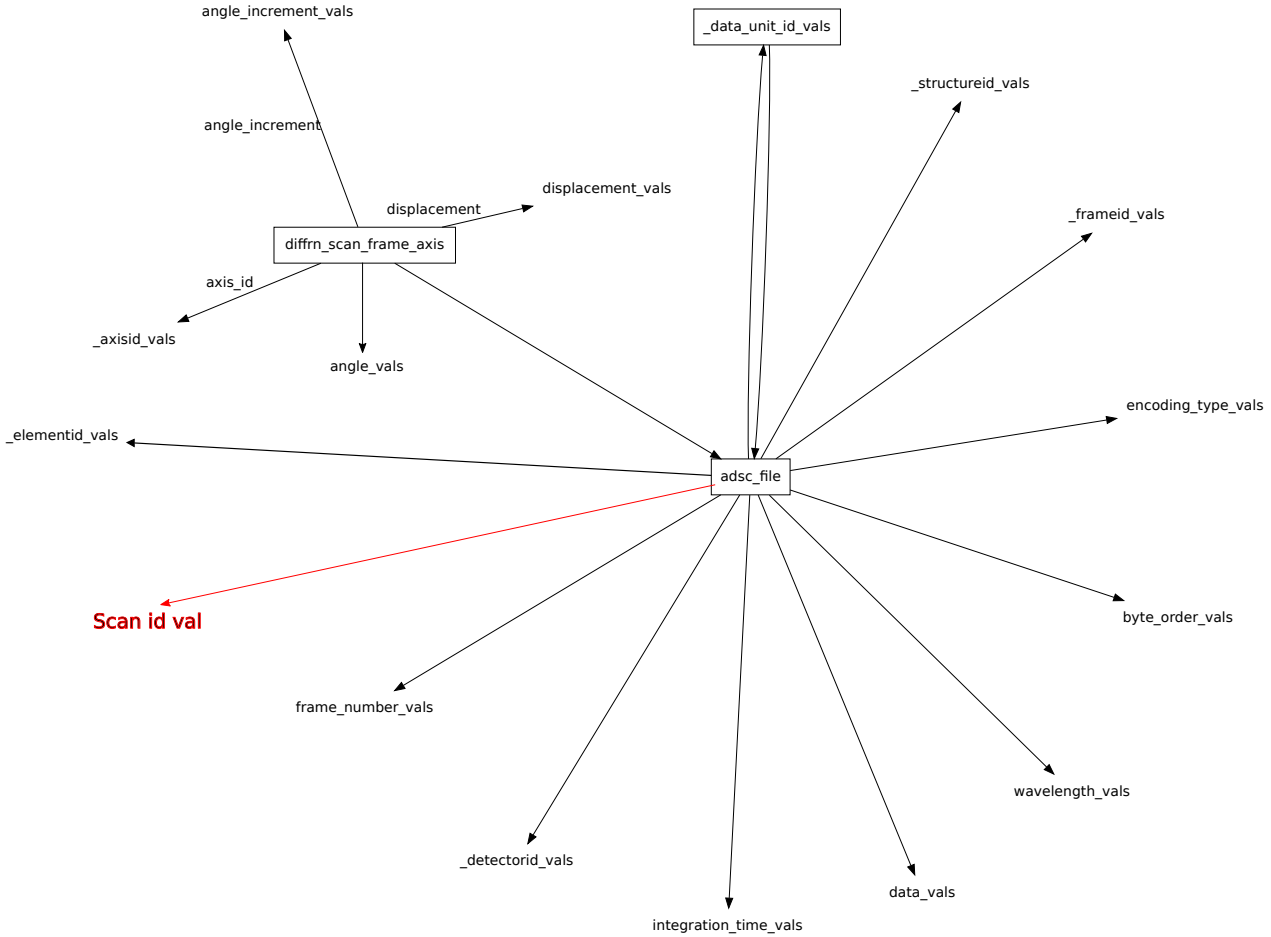
"Objects" connected by "morphisms"

A collection of tables (a relational database) = a category over sets and functions

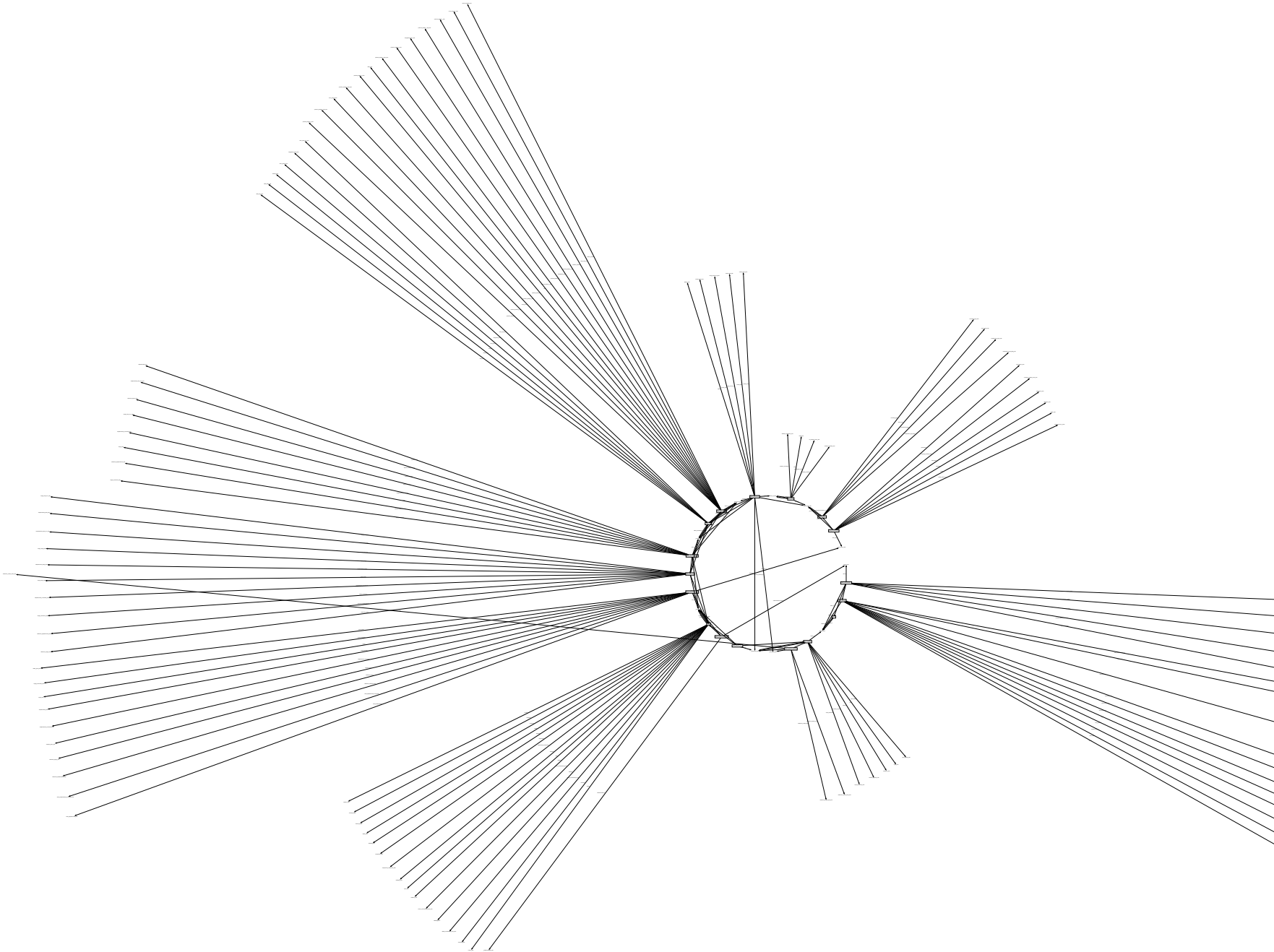
ADSC file collection as a category



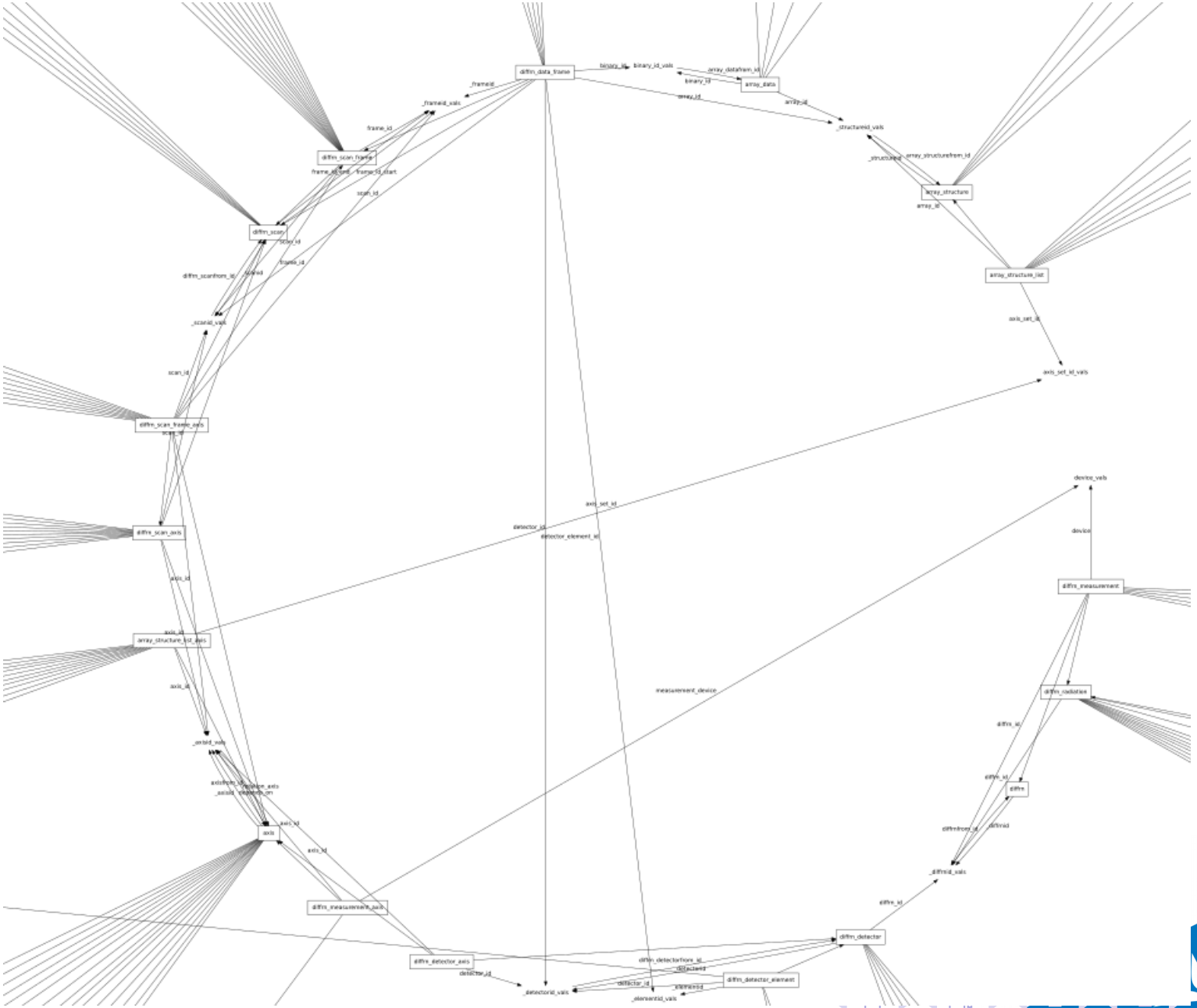
Grouping into scans



imgCIF diagrammed



imgCIF, zoomed in



Mappings between ontologies (Functors)

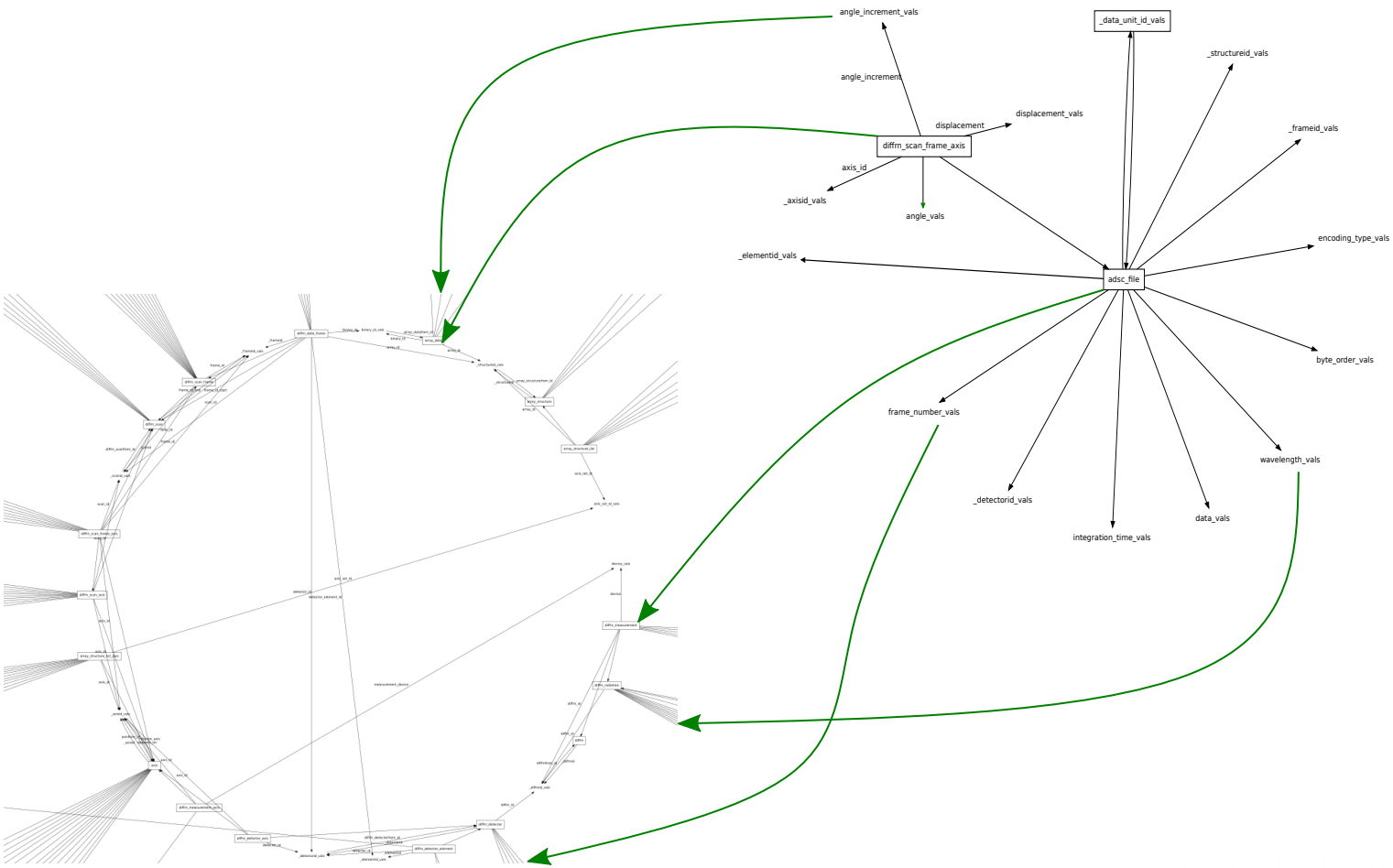
Objects go to objects, paths go to paths

Given a functor, automatic data migration between categories possible:

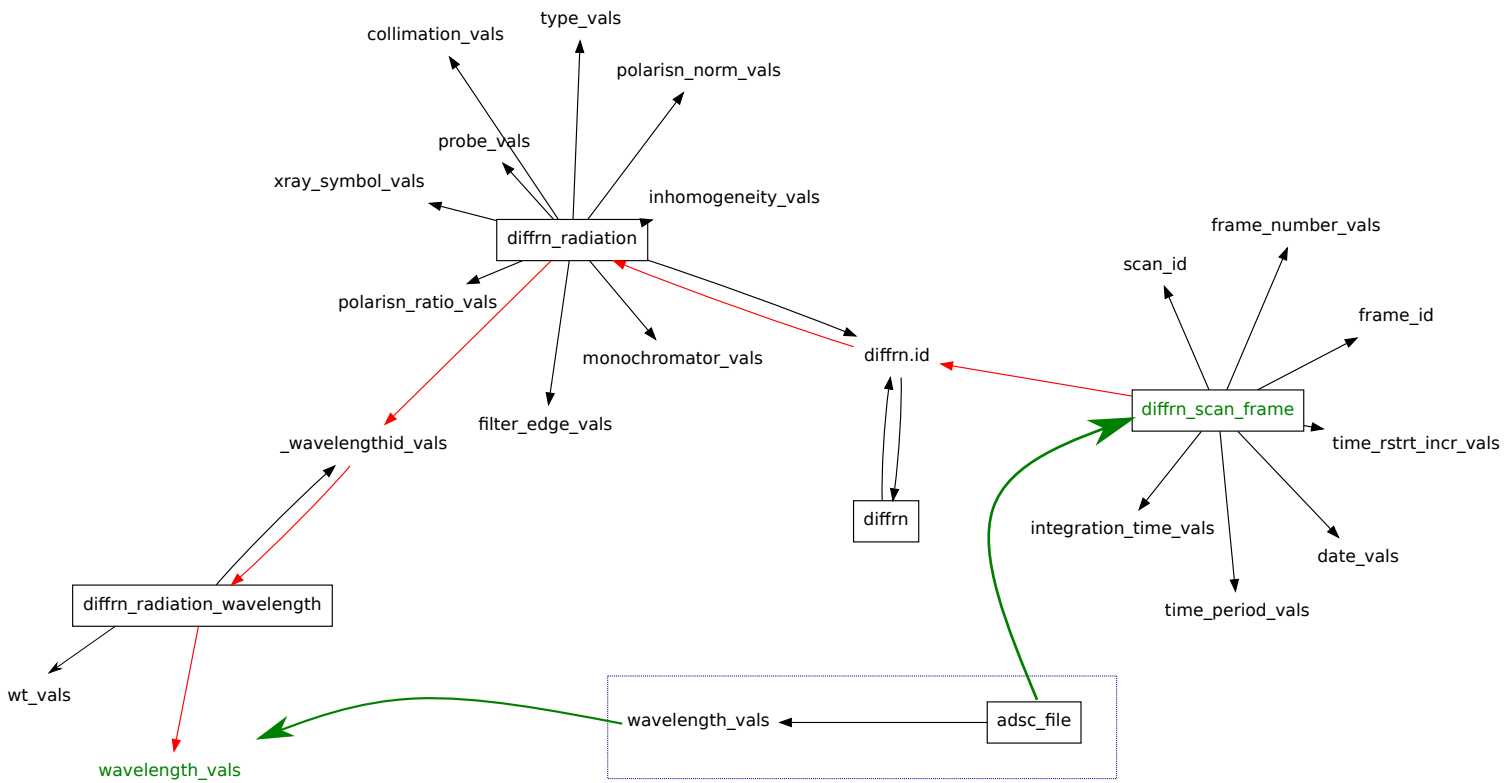
- Left and right pushforward functors
  - Equivalent to database joins and unions
- Machine-actionable
- Fills in the gaps

Spivak, D. I. "Functorial Data Migration" *Information and Computation* 217 (2012) 31-51
Useful also unpublished earlier version `http://math.mit.edu/~dspivak/informatics/FunctorialDataMigration.pdf`

Functor from ADSC category to imgCIF category



In detail


collimation_vals
type_vals
polarisn_norm_vals
probe_vals
xray_symbol_vals
inhomogeneity_vals
diffrn_radiation
polarisn_ratio_vals
monochromator_vals
filter_edge_vals
_wavelengthid_vals
diffrn.id
diffrn
frame_number_vals
scan_id
frame_id
diffrn_scan_frame
time_rstrt_incr_vals
integration_time_vals
date_vals
time_period_vals
diffrn_radiation_wavelength
wt_vals
wavelength_vals
wavelength_vals
adsc_file

Summary

- Tabular (categorical) representation is basis for interoperability standards
  - Machine-actionable descriptions of arbitrary formats
  - Links to community ontologies
- Need to specify only equivalences between tables and columns

A "virtual depository"

1. Provide one or more DOIs
2. System determines and lists file types found at those DOIs
3. Assign format specifiers to each file type

Later. . . .

1. Request a dataname
2. System returns values for this dataname
3. More sophisticated: return dataname corresponding to a given key

Outlook

- Proof of concept almost there
- Scaling up requires fast software
- Julia allows leveraging of existing C, Fortran libraries