DAMNIT: the Data And Metadata iNspection Interactive Thing

Thomas Michelat for the Data Analysis group

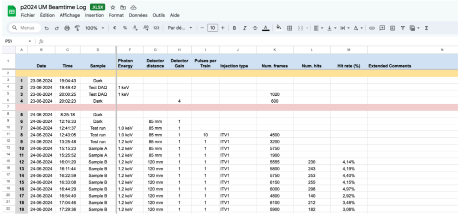
26 January 2024

What's DAMNIT?

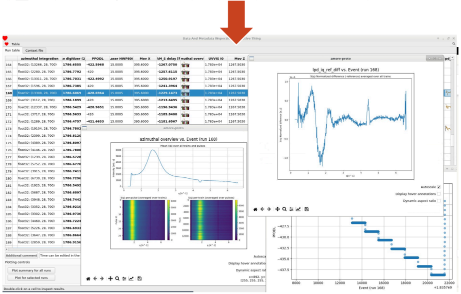
Original motivation: create run tables automatically.

Goals:

- Users often create spreadsheets with run data, we want to automate that.
- Lots of flexibility.
- Useful both during and after the beamtime.



	Date	Time	Sample	Photon Energy	Detector distance	Detector Gain	Pulses per Train	Injection type	Num. frames	Num. hits	Hit rate (%)	Extended Comments
1	23-06-2024	19:04:43	Dark									
2	23-06-2024	19:49:42	Test DAQ	1 keV								
3	23-06-2024	20:00:25	Test DAQ	1 keV					1020			
4	23-06-2024	20:02:23	Dark			4			600			
5	24-06-2024	8:25:18	Dark									
6	24-06-2024	12:16:33	Dark		85 mm	1						
7	24-06-2024	12:41:37	Test run	1.0 keV	85 mm	1						
8	24-06-2024	12:43:05	Test run	1.0 keV	85 mm	1	10	ITV1	4500			
9	24-06-2024	13:25:48	Test run	1.2 keV	85 mm	1	1	ITV1	3200			
10	24-06-2024	15:15:23	Sample A	1.2 keV	85 mm	1	1	ITV1	5750			
11	24-06-2024	15:25:52	Sample A	1.2 keV	85 mm	1	1	ITV1	1900			
12	24-06-2024	16:01:20	Sample B	1.2 keV	120 mm	1	1	ITV1	5555	230	4,14%	
13	24-06-2024	16:11:44	Sample B	1.2 keV	120 mm	1	1	ITV1	5800	243	4,19%	
14	24-06-2024	16:22:59	Sample B	1.2 keV	120 mm	1	1	ITV1	5750	253	4,40%	
15	24-06-2024	16:33:08	Sample B	1.2 keV	120 mm	1	1	ITV1	6150	255	4,15%	
16	24-06-2024	16:44:29	Sample B	1.2 keV	120 mm	1	1	ITV1	6000	298	4,97%	
17	24-06-2024	16:54:40	Sample B	1.2 keV	120 mm	1	1	ITV1	4800	140	2,92%	
18	24-06-2024	17:04:46	Sample B	1.2 keV	120 mm	1	1	ITV1	6100	212	3,48%	
19	24-06-2024	17:29:36	Sample B	1.2 keV	120 mm	1	1	ITV1	5900	182	3,08%	


Data And Metadata iNspecti... ...tive Thing
Table
Run table
Context file
amore-proto
azimuthal overview vs. Event (run 168)
Mean I(q) over all trains and pulses
lpd_iq_ref_diff vs. Event (run 168)
Autoscale
Display hover annotations
Dynamic aspect ratio
PPODL
Event (run 168)
Plotting controls
Plot summary for all runs
Plot for selected runs
Additional comment

	azimuthal integration	w digitizer (2	PPODL	aser HWP80	Mov X	M_S delay [f	muthal overv	UVVIS I0	Mov Z
164	float32: (13266, 28, 700)	1786.6555	-422.5968	15.0005	395.6000	-1267.0750		1.783e+04	1267.5030
165	float32: (2280, 28, 700)	1786.7792	-420	15.0005	395.6000	-1257.6115		1.783e+04	1267.5030
166	float32: (13311, 28, 700)	1786.7031	-422.4992	15.0005	395.6000	-1250.9197		1.783e+04	1267.5030
167	float32: (1596, 28, 700)	1786.7385	-420	15.0005	395.6000	-1241.3964		1.783e+04	1267.5030
168	float32: (13308, 28, 700)	1786.6069	-428.6964	15.0005	395.6000	-1229.1473		1.783e+04	1267.5030
169	float32: (3112, 28, 700)	1786.1899	-420	15.0005	395.6000	-1213.6495		1.783e+04	1267.5030
170	float32: (12337, 28, 700)	1786.5429	-420.9651	15.0005	395.6000	-1196.9436		1.783e+04	1267.5030
171	float32: (3717, 28, 700)	1786.5633	-420	15.0005	395.6000	-1185.8400		1.783e+04	1267.5030
172	float32: (1289, 28, 700)	1786.4757	-421.6633	15.0005	395.6000	-1181.6567		1.783e+04	1267.5030
173	float32: (19104, 28, 700)	1786.7502							
174	float32: (2399, 28, 700)	1786.8120							
175	float32: (4389, 28, 700)	1786.8097							
176	float32: (4146, 28, 700)	1786.7808							
177	float32: (1239, 28, 700)	1786.5728							
178	float32: (5752, 28, 700)	1786.6770							
179	float32: (3915, 28, 700)	1786.7411							
180	float32: (6730, 28, 700)	1786.7296							
181	float32: (1925, 28, 700)	1786.5492							
182	float32: (5687, 28, 700)	1786.6897							
183	float32: (3948, 28, 700)	1786.7442							
184	float32: (3352, 28, 700)	1786.9210							
185	float32: (3302, 28, 700)	1786.9736							
186	float32: (4460, 28, 700)	1786.7224							
187	float32: (5226, 28, 700)	1786.6933							
188	float32: (3647, 28, 700)	1786.8664							
189	float32: (2859, 28, 700)	1786.9156							

Time can be edited in the

Double-click on a cell to inspect results.

What’s DAMNIT?

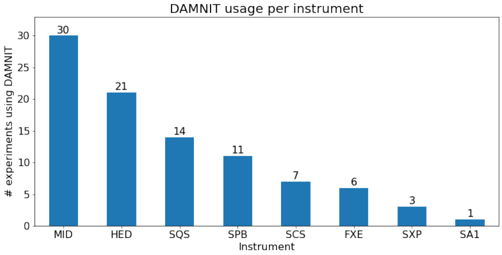
Original motivation: create run tables automatically.

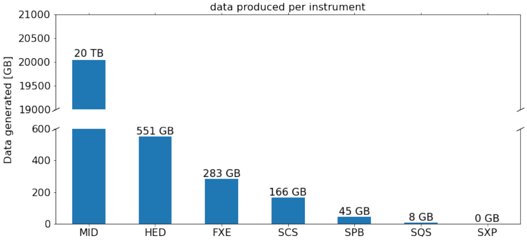
Goals:

- Users often create spreadsheets with run data, we want to automate that.
- Lots of flexibility.
- Useful both during and after the beamtime.

Status:

- Used in ∼85 proposals across all instruments.
- Generated at least ∼21TB of data.
- Analysis of e.g. motors, digitizers, and 2D detectors.
- Used to trigger external analysis tools.





Technical details

- Prototype is written in Python/PyQt, will move to a web frontend (currently in active development).
- Separate backend and frontend.

Technical details

- Prototype is written in Python/PyQt, will move to a web frontend (currently in active development).
- Separate backend and frontend.
- Backend executes a *context file* that contains `Variable`'s:

```python
from damnit_ctx import Variable

@Variable(title="R1 motor", summary="mean")
def R1(run):
    return run["MID_EXP_UPP/MOTOR/R1", "actualPosition"].xarray()
```

Technical details

- Prototype is written in Python/PyQt, will move to a web frontend (currently in active development).
- Separate backend and frontend.
- Backend executes a *context file* that contains `Variable`'s:

```python
from damnit_ctx import Variable

@Variable(title="R1 motor", summary="mean")
def R1(run):
    return run["MID_EXP_UPP/MOTOR/R1", "actualPosition"].xarray()
```

- What happens when a new run is taken:
  1. New data triggers DAMNIT processing
  2. Backend executes `Variable`'s, saves to HDF5 files and SQLite.
  3. Backend notifies frontends to update with Kafka.
  4. If used, calibration pipeline triggers backend again for `proc/` data.

Technical details

- HDF5 file created for each run.
- Contains both reduced and complete `Variable` data.

```
$ ls
context.py  extracted_data  runs.sqlite

$ h5glance extracted_data/p1234_r100.h5
p1234_r100.h5
├.reduced
│ ├adc_9_diode   [float64: scalar]
│ ├keithley_1    [float64: scalar]
│ ├n_trains      [int64: scalar]
│ ├pulses        [int64: scalar]
│ └xgm_intensity [float32: scalar]
├adc_9_diode
│ ├data          [float64: 1086] (3 attributes)
│ └trainId       [uint64: 1086] (4 attributes)
├keithley_1
│ ├data          [float64: 1086] (3 attributes)
│ └trainId       [uint64: 1086] (4 attributes)
├n_trains
│ └data          [int64: scalar]
├pulses
│ └data          [int64: scalar]
├xgm_intensity
│ ├data          [float32: 1086] (3 attributes)
│ └trainId       [uint64: 1086] (4 attributes)
└coor_plot
  └data          [uint8: 600 × 1300 × 4]
```

What's a Variable?

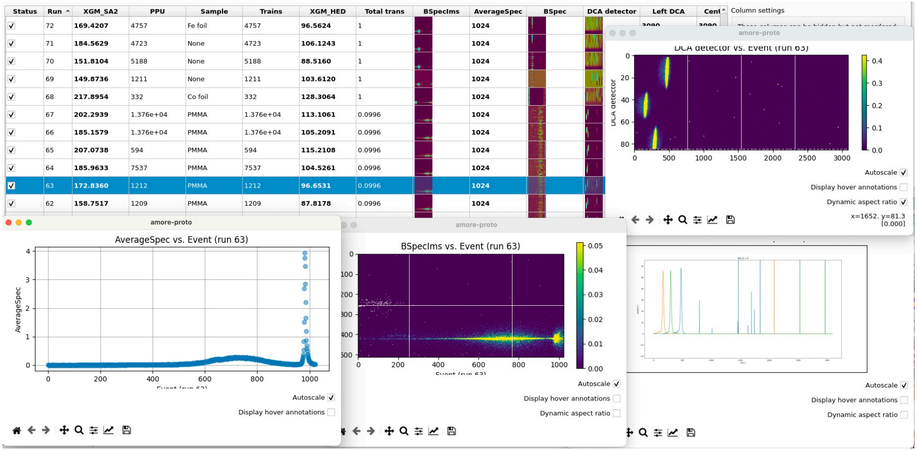
- Python decorator with attributes.
- Marks a function to be computed for each run.
- Can (currently) be a string, N-d array, scalar, or Figure.
- Represented by a column in the frontend.
- Can depend on each other.

```python
@Variable(title="Beam intensity", summary="mean")
def xgm_intensity(run):
    xgm_name = "SA2_XTD1_XGM/XGM/DOOCS:output"
    xgm = run[xgm_name, "data.intensityTD"].xarray()
    return xgm.sum(axis=1)

@Variable(title="AGIPD preview", data="proc", cluster=True)
def agipd_preview(run):
    agipd = AGIPD1M(run, min_modules=16)
    # Process data...
    geom = AGIPD_1MGeometry.from_crystfel_geom("/path/to/agipd.geom")
    image, center = geom.position_modules(processed_data)
    return image

@Variable(title="I(q)", summary="max")
def azimuthal_integration(run, image: "var#agipd_preview"):
    I, q = integrate1D(image)
    return xr.DataArray(I, dims=("intensity",), coords=dict(intensity=q))
```

DEMO

DEMO

Showcase - Inelastic x-ray scattering at HED (p003777 / Thomas Preston)



Status	Run	XGM_SA2	PPU	Sample	Trains	XGM_HED	Total trans	BSpecIms	AverageSpec	BSpec	DCA detector	Left DCA	Cent
☑	72	**169.4207**	4757	Fe foil	4757	**96.5624**	1		**1024**			3090	3090
☑	71	**184.5629**	4723	None	4723	**106.1243**	1		**1024**				
☑	70	**151.8104**	5188	None	5188	**88.5160**	1		**1024**				
☑	69	**149.8736**	1211	None	1211	**103.6120**	1		**1024**				
☑	68	**217.8954**	332	Co foil	332	**128.3064**	1		**1024**				
☑	67	**202.2939**	1.376e+04	PMMA	1.376e+04	**113.1061**	0.0996		**1024**				
☑	66	**185.1579**	1.376e+04	PMMA	1.376e+04	**105.2091**	0.0996		**1024**				
☑	65	**207.0738**	594	PMMA	594	**115.2108**	0.0996		**1024**				
☑	64	**185.9633**	7537	PMMA	7537	**104.5261**	0.0996		**1024**				
☑	63	**172.8360**	1212	PMMA	1212	**96.6531**	0.0996		**1024**				
☑	62	**158.7517**	1209	PMMA	1209	**87.8178**	0.0996		**1024**				

Column settings

Autoscale ☑

Display hover annotations □

Dynamic aspect ratio ☑

x=1652. y=81.3
[0.000]

Autoscale ☑

Display hover annotations □

Autoscale ☑

Display hover annotations □

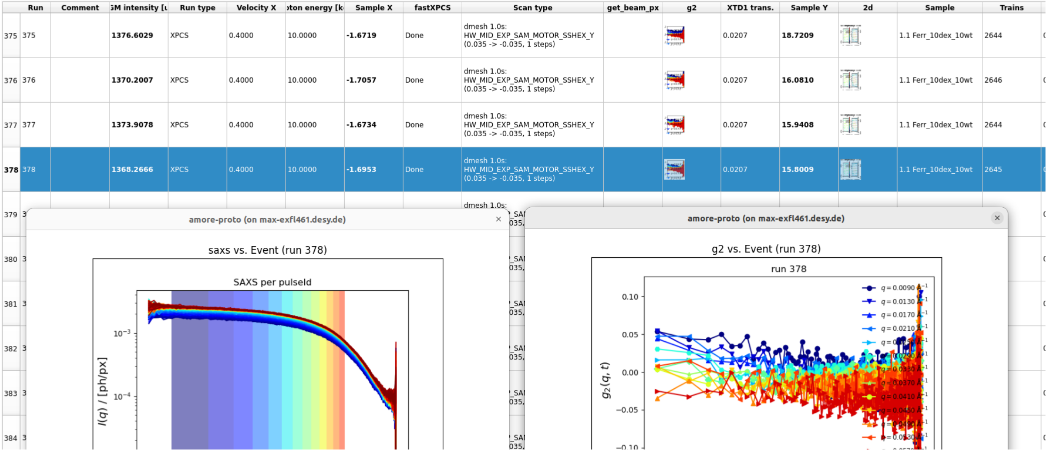
Dynamic aspect ratio □

Autoscale ☑

Display hover annotations □

Dynamic aspect ratio □

Showcase - MHz XPCS at MID (p005397 / Christian Gutt)

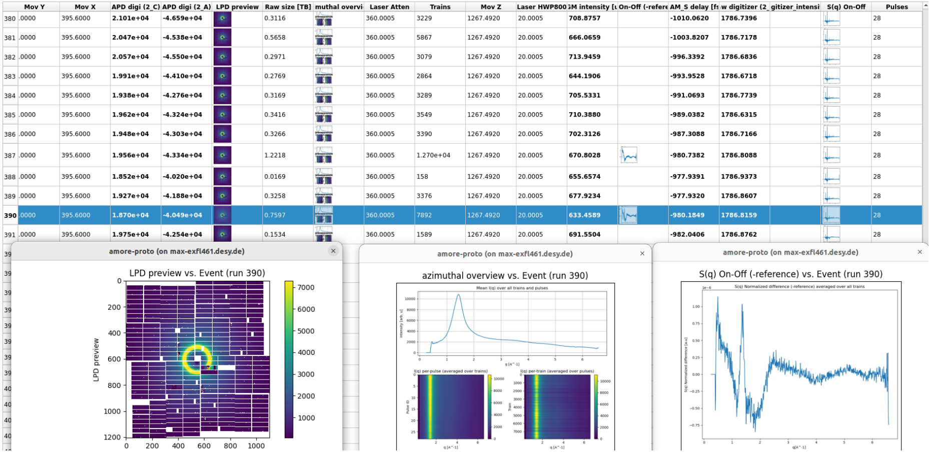


	Run	Comment	3M intensity [u	Run type	Velocity X	oton energy [k	Sample X	fastXPCS	Scan type	get_beam_px	g2	XTD1 trans.	Sample Y	2d	Sample	Trains
375	375		**1376.6029**	XPCS	0.4000	10.0000	**-1.6719**	Done	dmesh 1.0s: HW_MID_EXP_SAM_MOTOR_SSHEX_Y (0.035 -> -0.035, 1 steps)			0.0207	**18.7209**		1.1 Ferr_10dex_10wt	2644
376	376		**1370.2007**	XPCS	0.4000	10.0000	**-1.7057**	Done	dmesh 1.0s: HW_MID_EXP_SAM_MOTOR_SSHEX_Y (0.035 -> -0.035, 1 steps)			0.0207	**16.0810**		1.1 Ferr_10dex_10wt	2646
377	377		**1373.9078**	XPCS	0.4000	10.0000	**-1.6734**	Done	dmesh 1.0s: HW_MID_EXP_SAM_MOTOR_SSHEX_Y (0.035 -> -0.035, 1 steps)			0.0207	**15.9408**		1.1 Ferr_10dex_10wt	2644
378	378		**1368.2666**	XPCS	0.4000	10.0000	**-1.6953**	Done	dmesh 1.0s: HW_MID_EXP_SAM_MOTOR_SSHEX_Y (0.035 -> -0.035, 1 steps)			0.0207	**15.8009**		1.1 Ferr_10dex_10wt	2645
379	3								dmesh 1.0s: _SAI 035,							
380	3								_SAI 035,							
381	3								_SAI 035,							
382	3								_SAI 035,							
383	3								_SAI 035,							
384	3								_SAI 035,							

amore-proto (on max-exfl461.desy.de)

amore-proto (on max-exfl461.desy.de)

Showcase - HERFD-XAS and WAXS at FXE (p004507 / Christopher Milne)



	Mov Y	Mov X	APD digi (2_C)	APD digi (2_A)	LPD preview	Raw size [TB]	muthal overvi	Laser Atten	Trains	Mov Z	Laser HWP800	GM intensity [u	On-Off (-refere	AM_S delay [fs	w digitizer (2_	gitizer_intensi	S(q) On-Off	Pulses
380	.0000	395.6000	2.101e+04	-4.659e+04		0.3116		360.0005	3229	1267.4920	20.0005	708.8757		-1010.0620	1786.7396			28
381	.0000	395.6000	2.047e+04	-4.538e+04		0.5658		360.0005	5867	1267.4920	20.0005	666.0659		-1003.8207	1786.7178			28
382	.0000	395.6000	2.057e+04	-4.550e+04		0.2971		360.0005	3079	1267.4920	20.0005	713.9459		-996.3392	1786.6836			28
383	.0000	395.6000	1.991e+04	-4.410e+04		0.2769		360.0005	2864	1267.4920	20.0005	644.1906		-993.9528	1786.6718			28
384	.0000	395.6000	1.938e+04	-4.276e+04		0.3169		360.0005	3289	1267.4920	20.0005	705.5331		-991.0693	1786.7739			28
385	.0000	395.6000	1.962e+04	-4.324e+04		0.3416		360.0005	3549	1267.4920	20.0005	710.3880		-989.0382	1786.6315			28
386	.0000	395.6000	1.948e+04	-4.303e+04		0.3266		360.0005	3390	1267.4920	20.0005	702.3126		-987.3088	1786.7166			28
387	.0000	395.6000	1.956e+04	-4.334e+04		1.2218		360.0005	1.270e+04	1267.4920	20.0005	670.8028		-980.7382	1786.8088			28
388	.0000	395.6000	1.852e+04	-4.020e+04		0.0169		360.0005	158	1267.4920	20.0005	655.6574		-977.9391	1786.9373			28
389	.0000	395.6000	1.927e+04	-4.188e+04		0.3258		360.0005	3376	1267.4920	20.0005	677.9234		-977.9320	1786.8607			28
390	.0000	395.6000	1.870e+04	-4.049e+04		0.7597		360.0005	7892	1267.4920	20.0005	633.4589		-980.1849	1786.8159			28
391	.0000	395.6000	1.975e+04	-4.254e+04		0.1534		360.0005	1589	1267.4920	20.0005	691.5504		-982.0406	1786.8762			28

amore-proto (on max-exfl461.desy.de)

LPD preview vs. Event (run 390)

amore-proto (on max-exfl461.desy.de)

azimuthal overview vs. Event (run 390)

Mean I(q) over all trains and pulses
Intensity [arb. u]
q [A^-1]
I(q) per-pulse (averaged over trains)
Pulse ID
I(q) per-train (averaged over pulses)
Train
q [A^-1]

amore-proto (on max-exfl461.desy.de)

S(q) On-Off (-reference) vs. Event (run 390)

S(q) Normalized difference (-reference) averaged over all trains
1e-6
S(q) Normalized difference [a.u]
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
q[A^-1]

Future plans

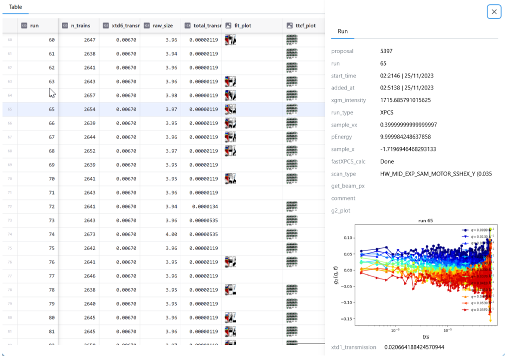
We have lots, but the most important ones are:

- Replace the current PyQt GUI with a web frontend
- Make a Python API for users to access `Variables`
- Parameters
- Filtering and grouping of runs and `Variables`
- Reproducibility (e.g. storing code and history of generated `Variable`)
- Sample table

Future plans

We have lots, but the most important ones are:

- Replace the current PyQt GUI with a web frontend
- Make a Python API for users to access `Variables`
- Parameters
- Filtering and grouping of runs and `Variables`
- Reproducibility (e.g. storing code and history of generated `Variable`)
- Sample table



	run	n_trains	xtd6_transr	raw_size	total_transr	fit_plot	ttcf_plot
60	60	2647	0.00670	3.96	0.00000119		
61	61	2638	0.00670	3.94	0.00000119		
62	62	2641	0.00670	3.96	0.00000119		
63	63	2643	0.00670	3.96	0.00000119		
64	[illegible]	2657	0.00670	3.98	0.00000119		
65	65	2654	0.00670	3.97	0.00000119		
66	66	2639	0.00670	3.95	0.00000119		
67	67	2644	0.00670	3.96	0.00000119		
68	68	2652	0.00670	3.97	0.00000119		
69	69	2639	0.00670	3.95	0.00000119		
70	70	2641	0.00670	3.95	0.00000119		
71	71	2643	0.00670	3.96	0.00000119		
72	72	2641	0.00670	3.94	0.0000134		
73	73	2643	0.00670	3.96	0.00000535		
74	74	2673	0.00670	4.00	0.00000535		
75	75	2642	0.00670	3.96	0.00000119		
76	76	2641	0.00670	3.95	0.00000119		
77	77	2646	0.00670	3.96	0.00000119		
78	78	2638	0.00670	3.95	0.00000119		
79	79	2640	0.00670	3.95	0.00000119		
80	80	2645	0.00670	3.96	0.00000119		
81	81	2645	0.00670	3.96	0.00000119		

proposal	5397
run	65
start_time	02:2146 | 25/11/2023
added_at	02:5138 | 25/11/2023
xgm_intensity	1715.685791015625
run_type	XPCS
sample_vx	0.39999999999999997
pEnergy	9.999984248637858
sample_x	-1.7196946468293133
fastXPCS_calc	Done
scan_type	HW_MID_EXP_SAM_MOTOR_SSHEX_Y (0.035
get_beam_px	
comment	
g2_plot	
xtd1_transmission	0.020664188424570944

Summary

DAMNIT currently offers:

- A processing framework
  - That executes user-defined code aiming for maximum flexibility
  - Built on top of our analysis software ecosystem
- A prototype frontend
  - That shows a run table
  - That lets the user explore `Variable`'s
  - Soon to be replaced by a web application
- Usage across all instruments for everything from simple metadata collection to full-blown analysis.

Summary

DAMNIT currently offers:

- A processing framework
  - That executes user-defined code aiming for maximum flexibility
  - Built on top of our analysis software ecosystem
- A prototype frontend
  - That shows a run table
  - That lets the user explore `Variable`'s
  - Soon to be replaced by a web application
- Usage across all instruments for everything from simple metadata collection to full-blown analysis.

- Open source: `https://github.com/European-XFEL/DAMNIT`
- Documentation: `https://damnit.rtfd.io`
- contact: da@xfel.eu