das

A Python Framework for DAS

The ultimate use case:

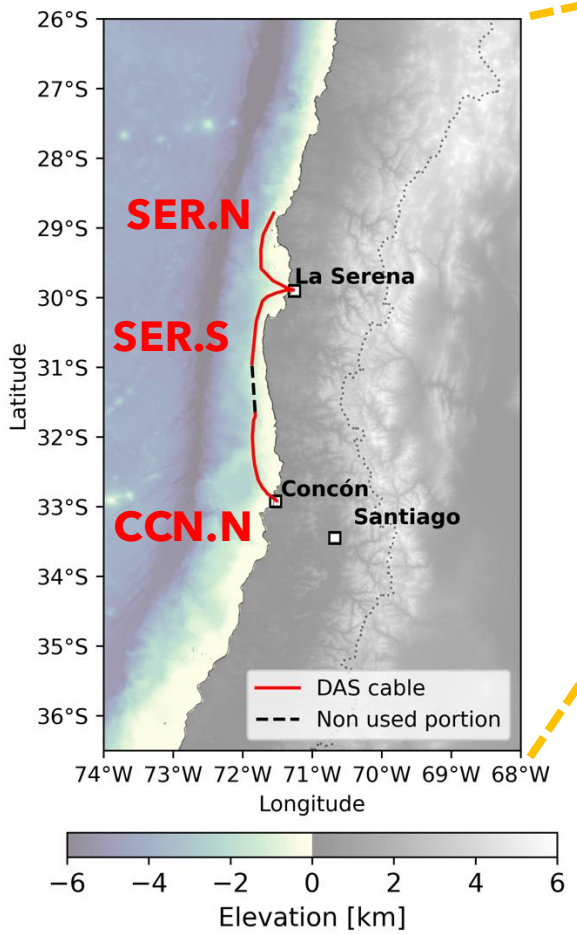
ABYSS
Monitoring megathrust faults with abyssal fiber optic cables

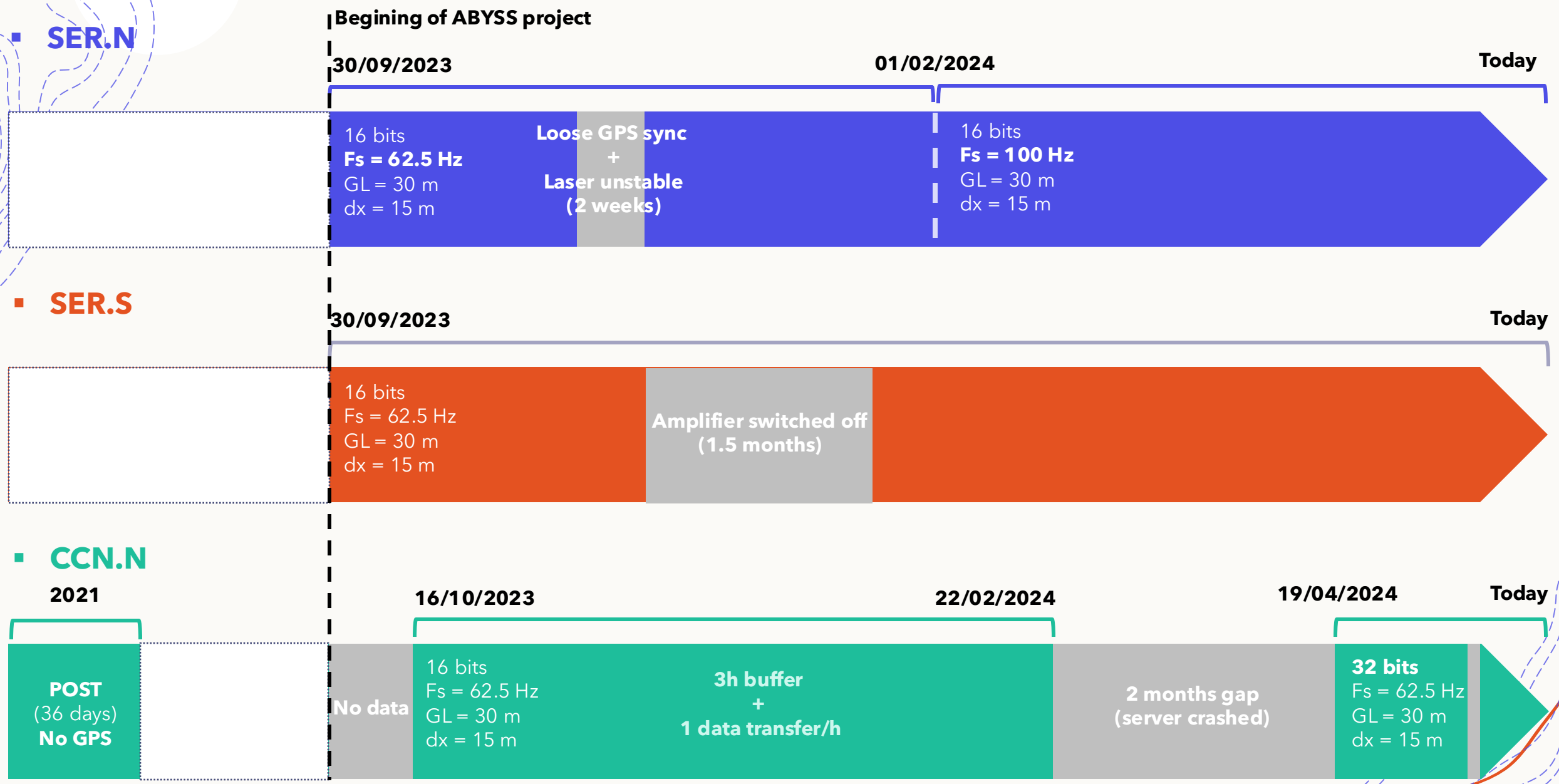
3 interrogators

450 km / 30 000 sensors

400 GB / day
25 000 files / day

5 years




SER.N
Begining of ABYSS project
30/09/2023
01/02/2024
Today
16 bits
Fs = 62.5 Hz
GL = 30 m
dx = 15 m
Loose GPS sync
+
Laser unstable
(2 weeks)
16 bits
Fs = 100 Hz
GL = 30 m
dx = 15 m
SER.S
30/09/2023
Today
16 bits
Fs = 62.5 Hz
GL = 30 m
dx = 15 m
Amplifier switched off
(1.5 months)
CCN.N
2021
16/10/2023
22/02/2024
19/04/2024
Today
POST
(36 days)
No GPS
No data
16 bits
Fs = 62.5 Hz
GL = 30 m
dx = 15 m
3h buffer
+
1 data transfer/h
2 months gap
(server crashed)
32 bits
Fs = 62.5 Hz
GL = 30 m
dx = 15 m

Key Features of 

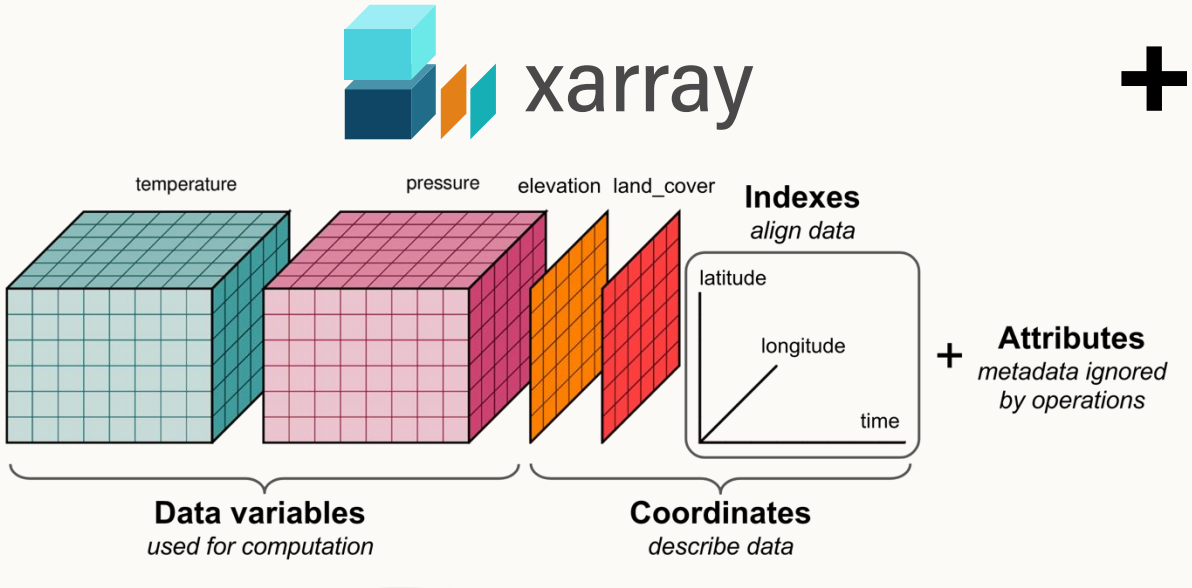
+ Seamless manipulation of any multi-file DAS datasets
+ Larger-than-memory computing (online processing)
+ Easy and Extensible (similar to Numpy/Scipy/Xarray)
+ Efficient signal processing routines (multi-threading)
+ Real-time capabilities

Xdas workflow

Xdas workflow

Sources of Inspiration

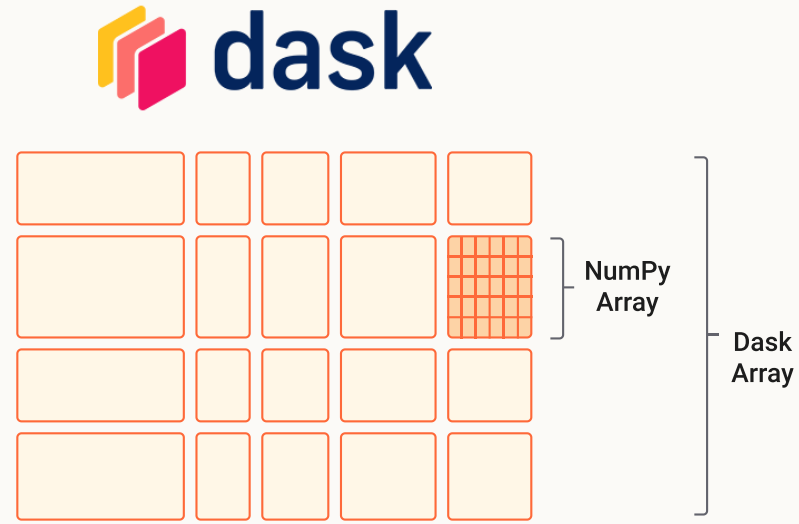
N-D labeled arrays (DataArray)



Only support **dense** coordinates

+

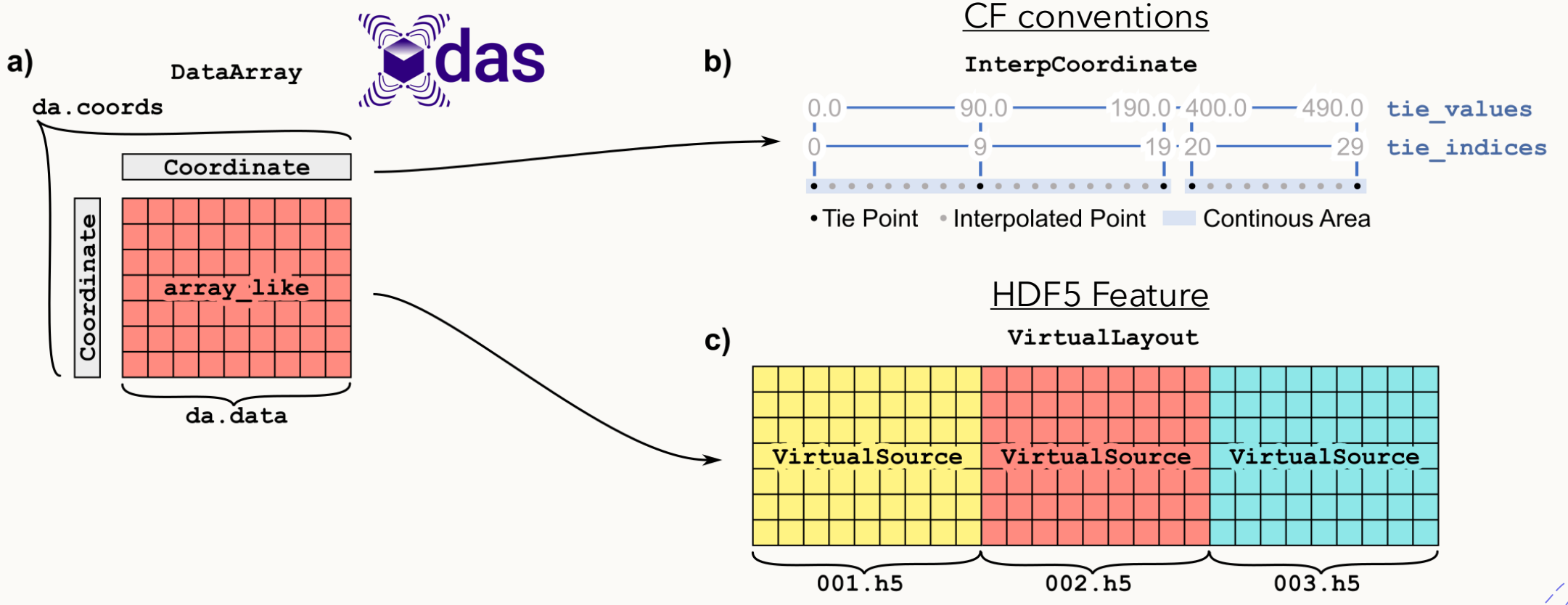
Chunked processing



One task object per chunk make **enormous redundant graphs**

➔ Reimplement a subset of Xarray with **interpolated** coordinates and **virtual** dataset

Data structure: DataArray



Code Example: Link files

```python
import xdas as xd

# link virtual dataset
da = xd.open_mfdataarray("abyss/CCN/N/20240101/*.hdf5", engine="asn", tolerance=None)
```

```
<xdas.DataArray (time: 5400000, distance: 10000)>
VirtualStack: 100.6G (int16)
Coordinates:
  * time (time): 2024-01-22T00:00:07.000 to 2024-01-23T00:00:06.984
  * distance (distance): 0.0 to 153179.709
```

```python
# save consolidation
da.to_netcdf("da.nc")
```

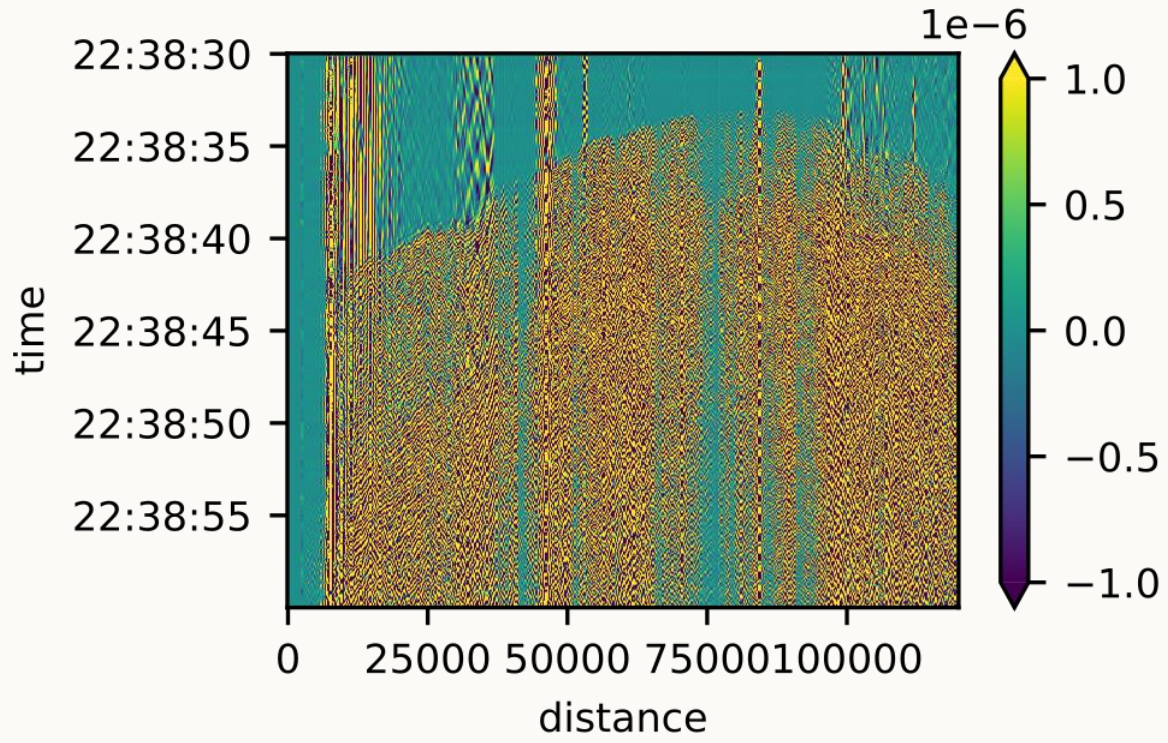
Code Example: Label based selection

```python
da = da.sel(
    time=slice("2024-01-22T22:38:30", "2024-01-22T22:39:00"),
    distance=slice(None, 120_000),
)
```

```
<xdas.DataArray (time: 1875, distance: 7834)>
VirtualStack: 28.0M (int16)
Coordinates:
  * time (time): 2024-01-22T22:38:30.008 to 2024-01-23T22:38:30.992
  * distance (distance): 0.0 to 119997.6662
```

Code Example: Plotting

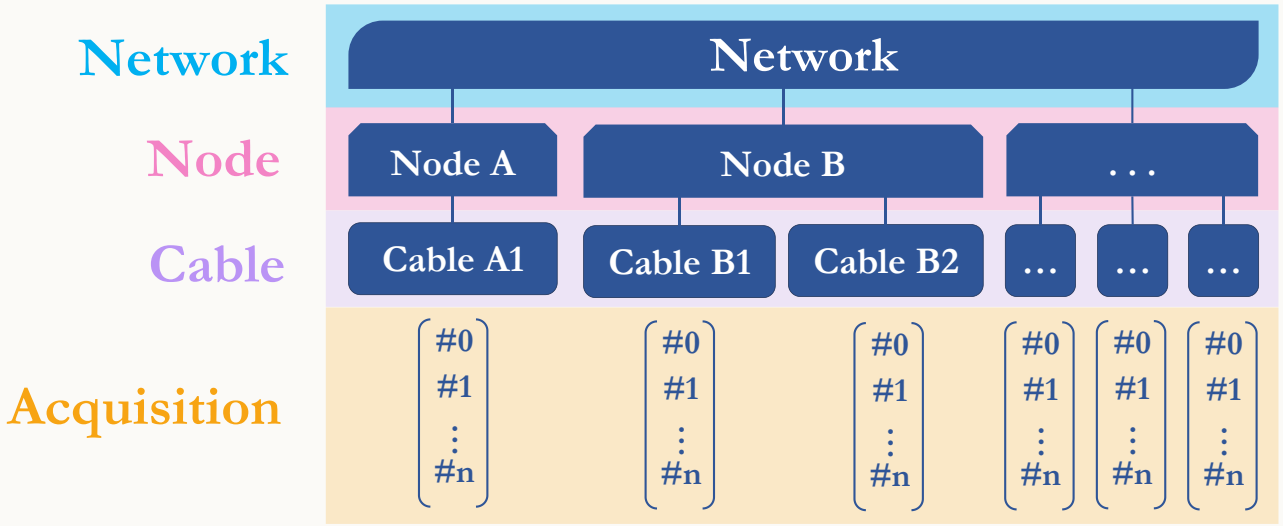
```
da.plot(yincrease=False, vmin=-1e-6, vmax=1e-6)
```



Data structure: DataCollection

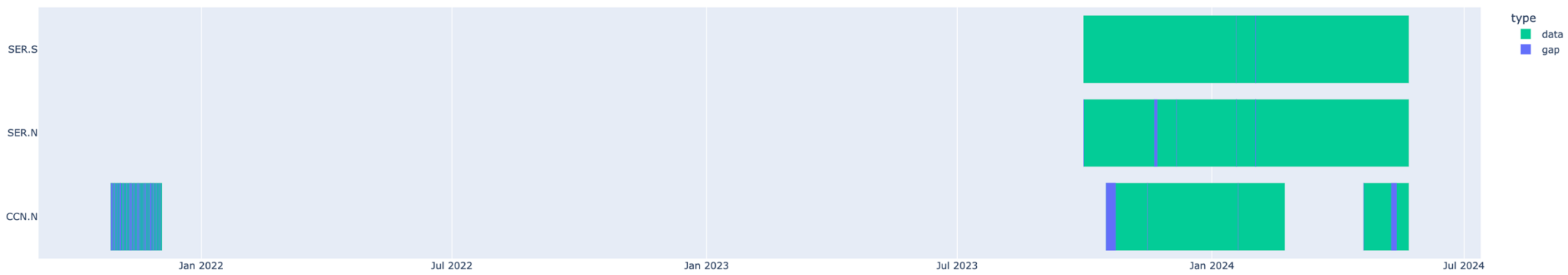
Hierarchy of DataArray objects

+ Multiple instruments
+ Catalog of events
+ Inhomogeneous acquisitions



Code Example: data availability

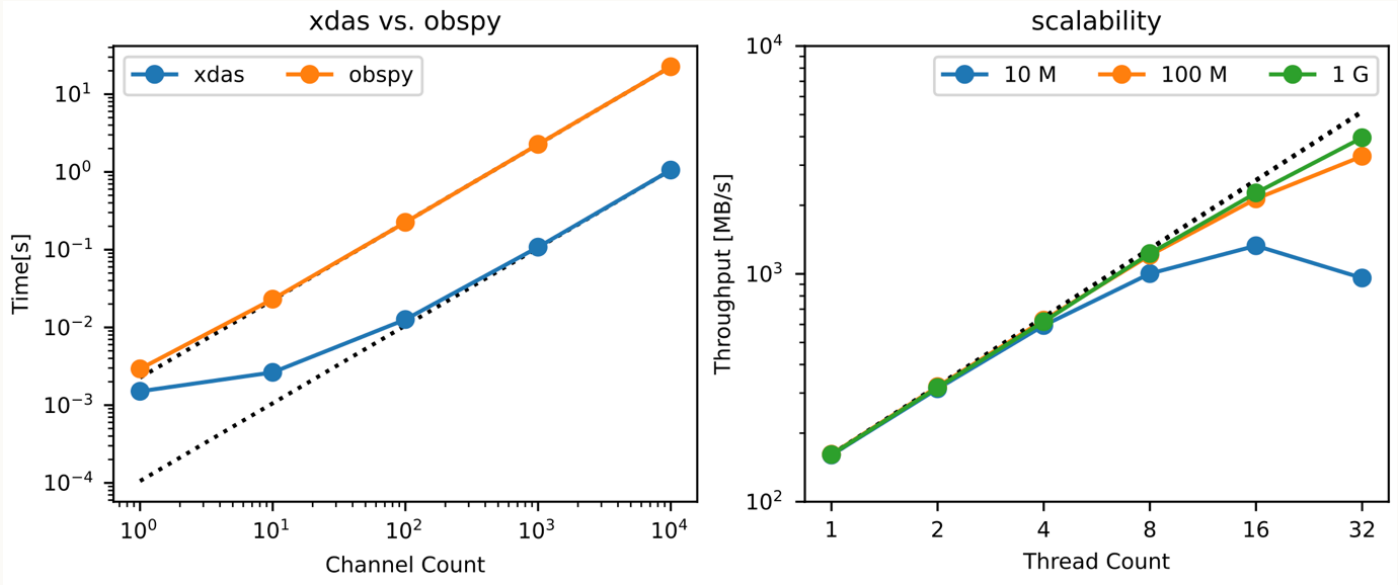
```
import xdas as xd

dc = xd.open_datacollection("abyss.nc")
xd.plot_availability(dc)
```



Xdas workflow

Multi-threaded signal processing routines

+ Implemented: scipy.signal and scipy.fft routines
+ Target: Single machine servers
+ Strategy: split & join



100x faster than obspy*

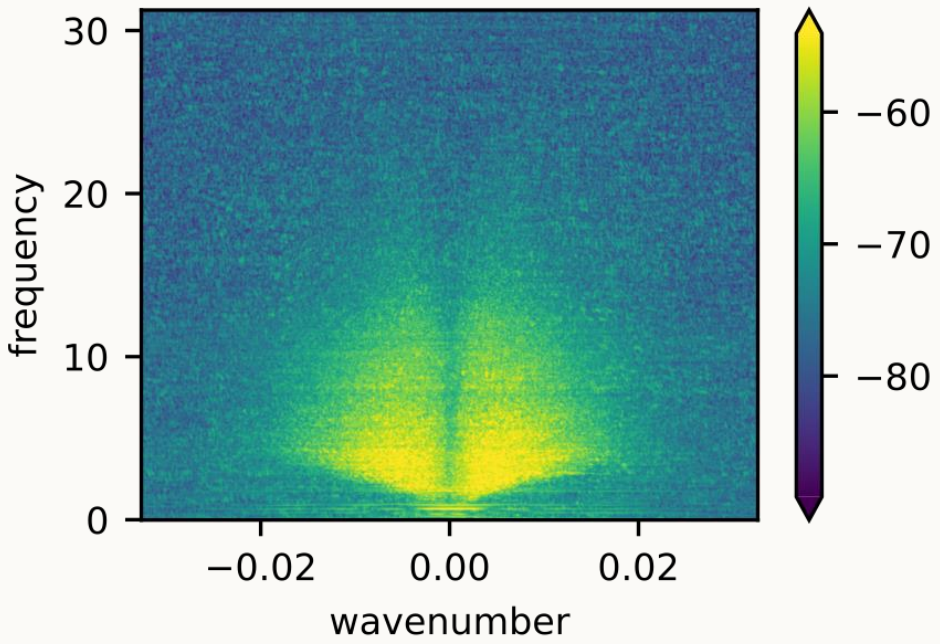
10x faster than scipy*

Code Example: signal processing

```python
import xdas.signal as xs

# apply gain
da *= 8.36e-11

# convert to velocity
da = xs.integrate(da, dim="distance")
da = xs.sliding_mean_removal(da, wlen=1000.0)

# decimate in time and space and plot
da = xs.decimate(da, 16, ftype="fir", dim="distance")
da = xs.decimate(da, 4, ftype="iir", dim="time")
```

Code Example: FFT processing

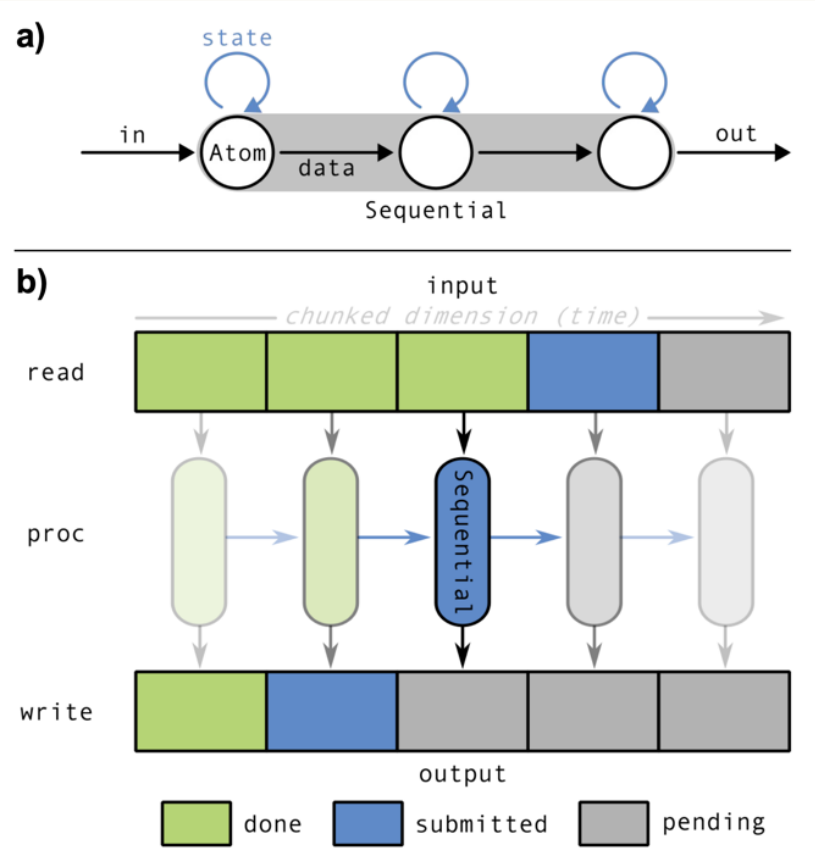
```python
import xdas.fft as xfft

# compute FK representation
fk = xs.taper(da, dim="distance")
fk = xs.taper(fk, dim="time")
fk = xfft.rfft(fk, dim={"time": "frequency"})
fk = xfft.fft(fk, dim={"distance": "wavenumber"})
fk = 20 * np.log10(np.abs(fk))  # numpy ufuncs

# plot
fk.plot(robust=True, interpolation="antialiased")
```



Xdas workflow

Online processing

+ Strategy:
  Pipes & Filters
+ Application:
  Larger-that-memory &
  Real-time
+ Model:
  Atom objects with state
  memory/passing



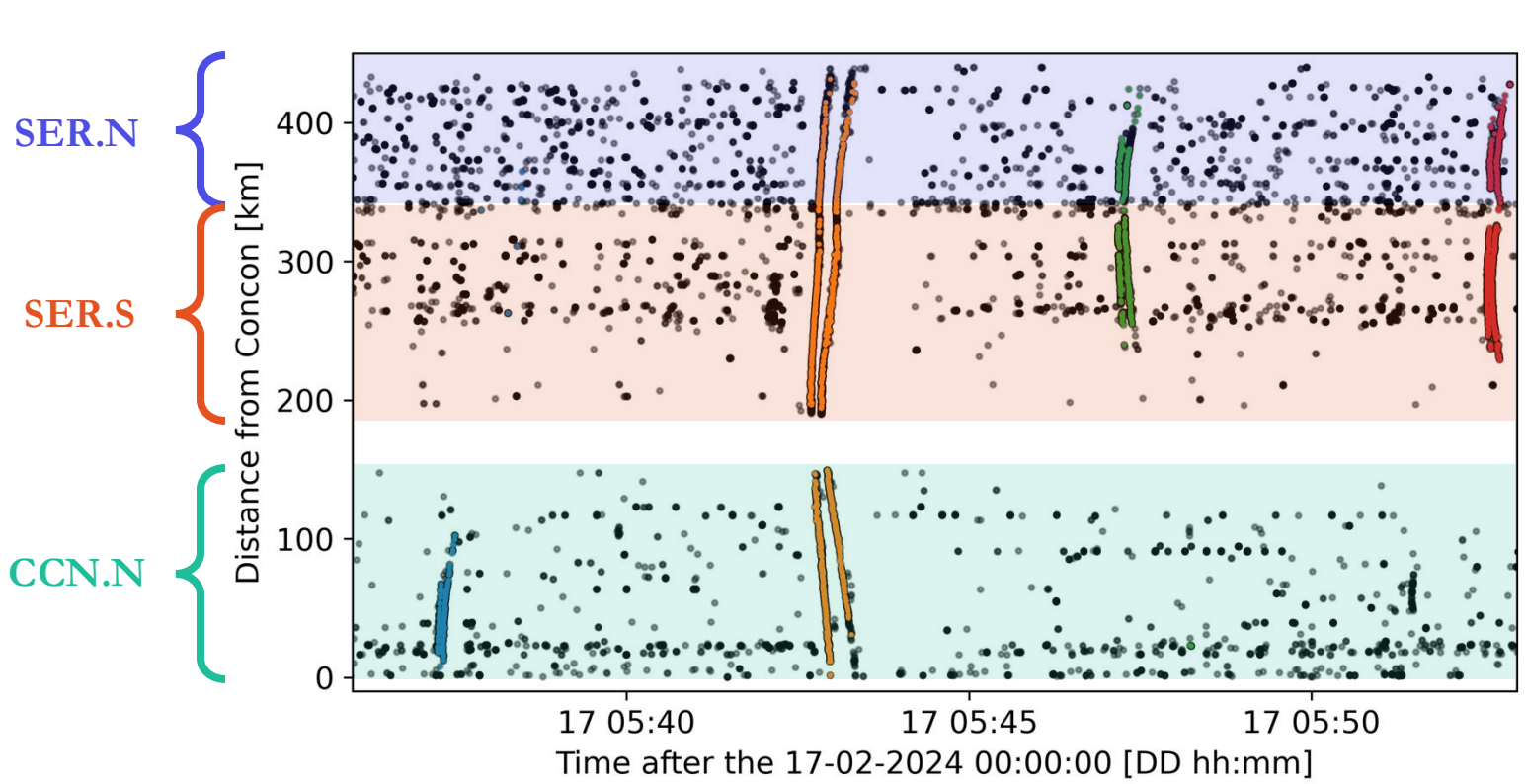
Code Example: Atoms

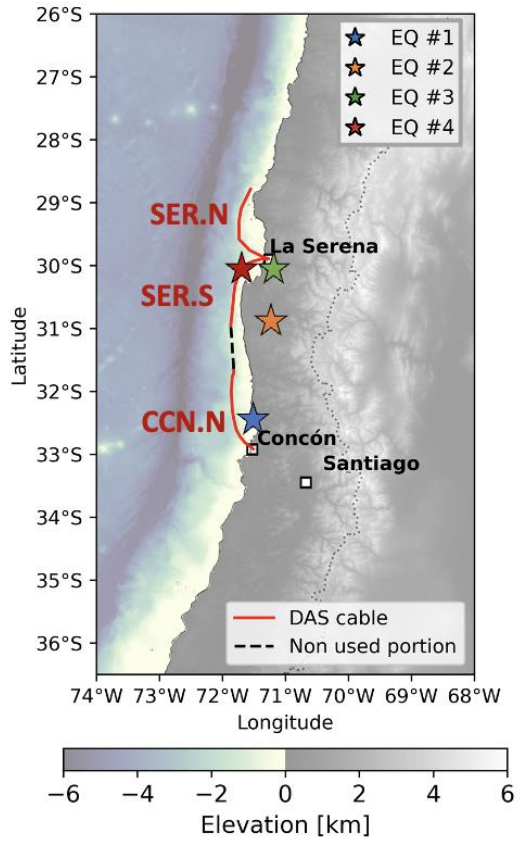
```python
import xdas.atoms as xa

atom = xa.Sequential(
    [
        xs.taper(..., dim="distance"),  # replace input by ...
        xa.DownSample(2, dim="time"),  # use stateful atoms
        np.square,  # already unary in/out operation
    ]
)

result = atom(da)
```

Code Example: Massive processing

```
import xdas.processing as xp

dl = xp.DataArrayLoader(da, chunks={"time": 10_000})
dw = xp.DataArrayWriter(path="outputs/")

result = xp.process(atom, dl, dw)
```

Use case: ML picking





https://pisco.unice.fr/#/

ABYSS Monitoring megathrust faults with abyssal fiber optic cables

FR EN

- Home
- Catalog
- Stations
- Live

Events in the study area (last 7 days)
Event time in UTC

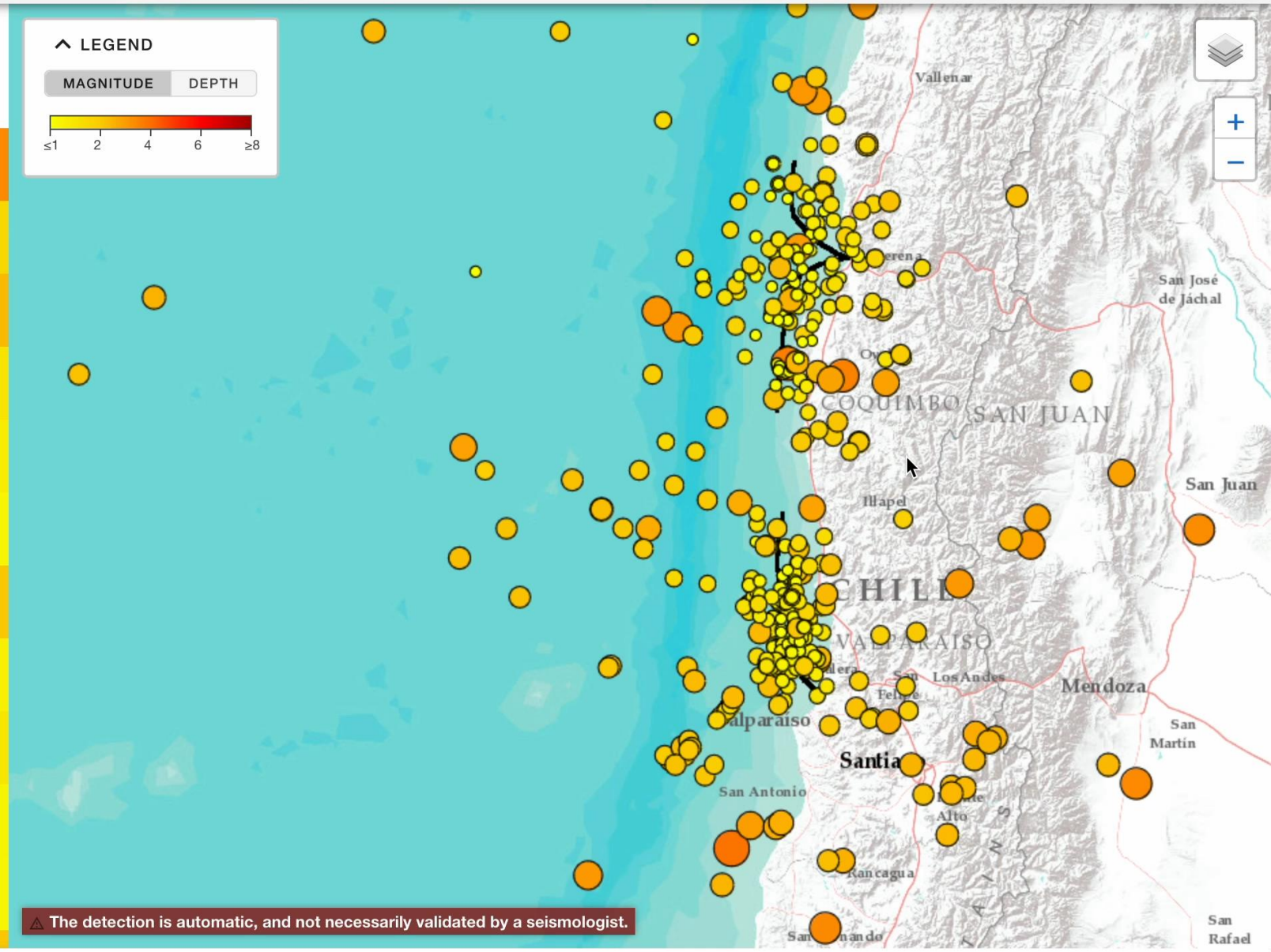
MAG

3.4	2024-10-10 23:57:07 (14h 5...)	MENDOZA, ARGENTINA
1.7	2024-10-10 23:33:12 (15h 1...)	OFF THE COAST OF CENTR...
2.4	2024-10-10 19:55:56 (18h 5...)	NEAR THE COAST OF CENT...
1.5	2024-10-10 15:23:37 (23h 2...)	NEAR THE COAST OF CENT...
1.3	2024-10-10 15:07:02 (23h 4...)	OFF THE COAST OF CENTR...
1.2	2024-10-10 15:05:22 (23h 4...)	NEAR THE COAST OF CENT...
2.1	2024-10-10 15:03:56 (23h 4...)	OFF THE COAST OF CENTR...
1.3	2024-10-10 14:39:41 (1j 11m)	NEAR THE COAST OF CENT...
1.5	2024-10-10 14:22:00 (1j 28m)	NEAR THE COAST OF CENT...
1.2	2024-10-10 14:15:14 (1j 35m)	NEAR THE COAST OF CENT...
1.2	2024-10-10 13:26:04 (1j 1h ...)	NEAR THE COAST OF CENT...

LEGEND

MAGNITUDE DEPTH

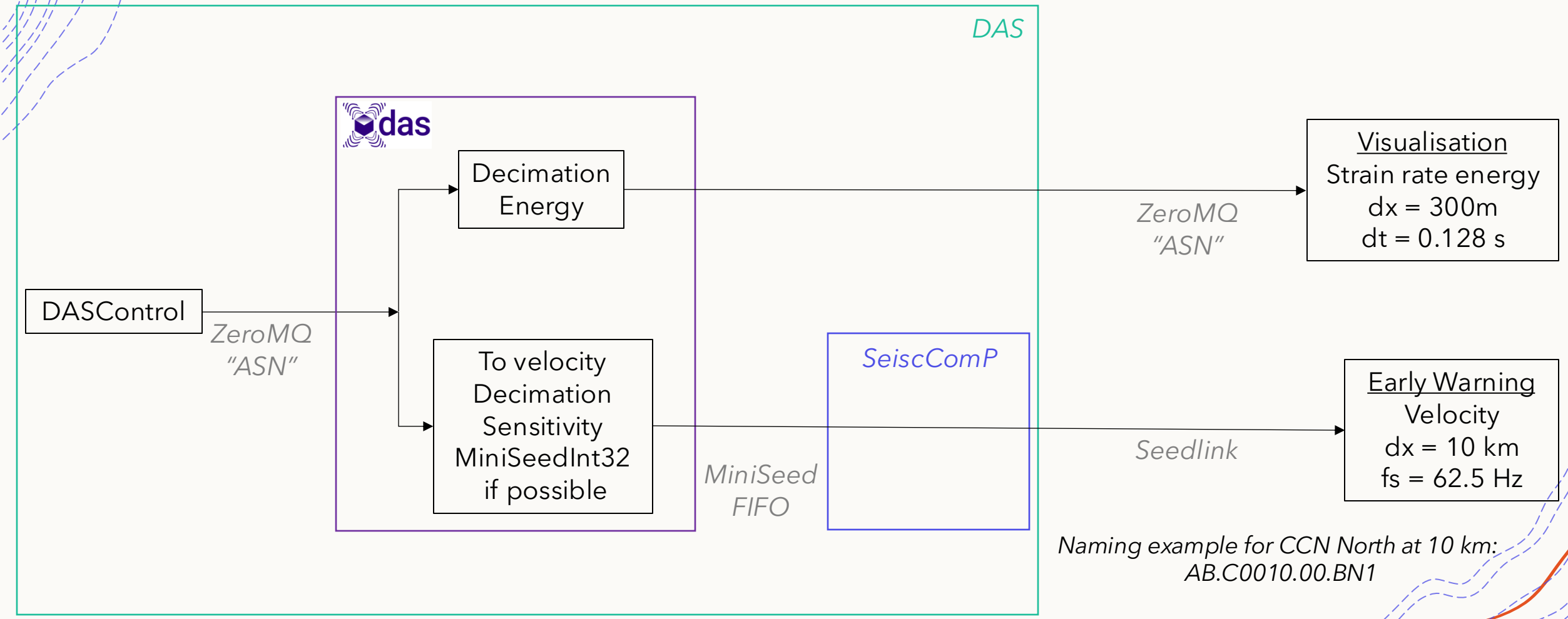
≤1 2 4 6 ≥8



⚠ The detection is automatic, and not necessarily validated by a seismologist.

Further information

Real-time streaming

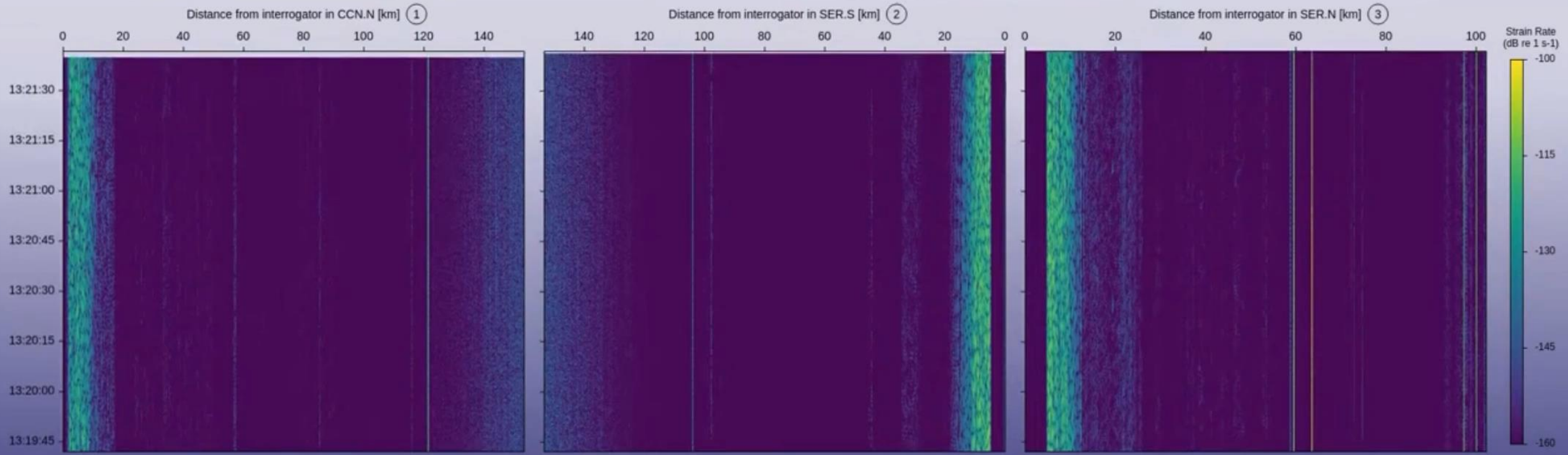


Naming example for CCN North at 10 km:
AB.C0010.00.BN1

ABYSS

Monitoring megathrust faults with abyssal fiber optic cables

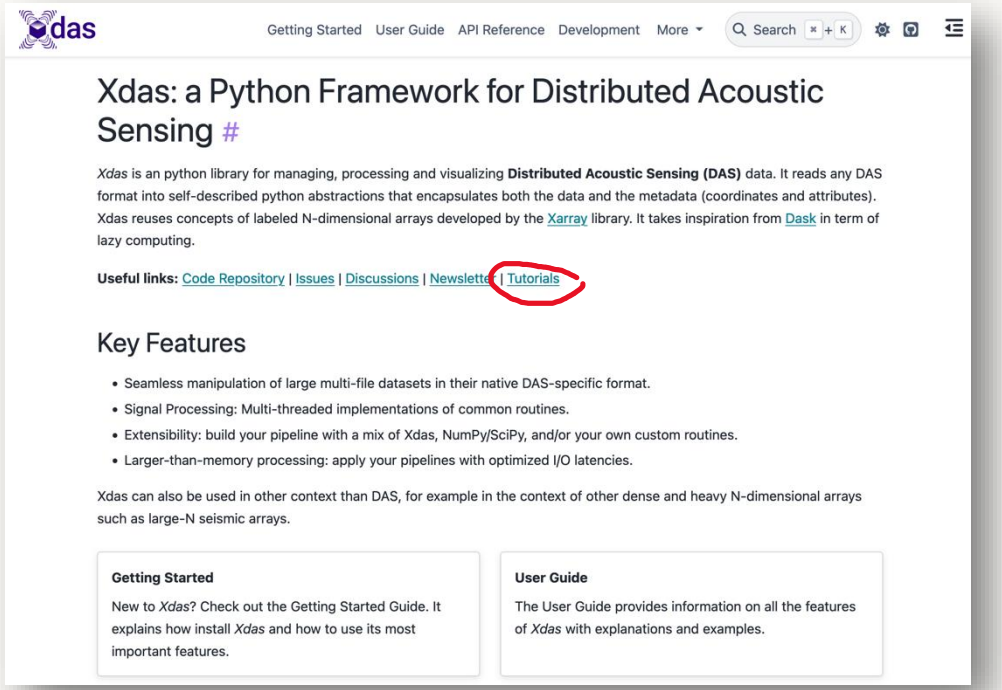
Live stream of seismic movement in **central Chile** along submarine fiber optic cables



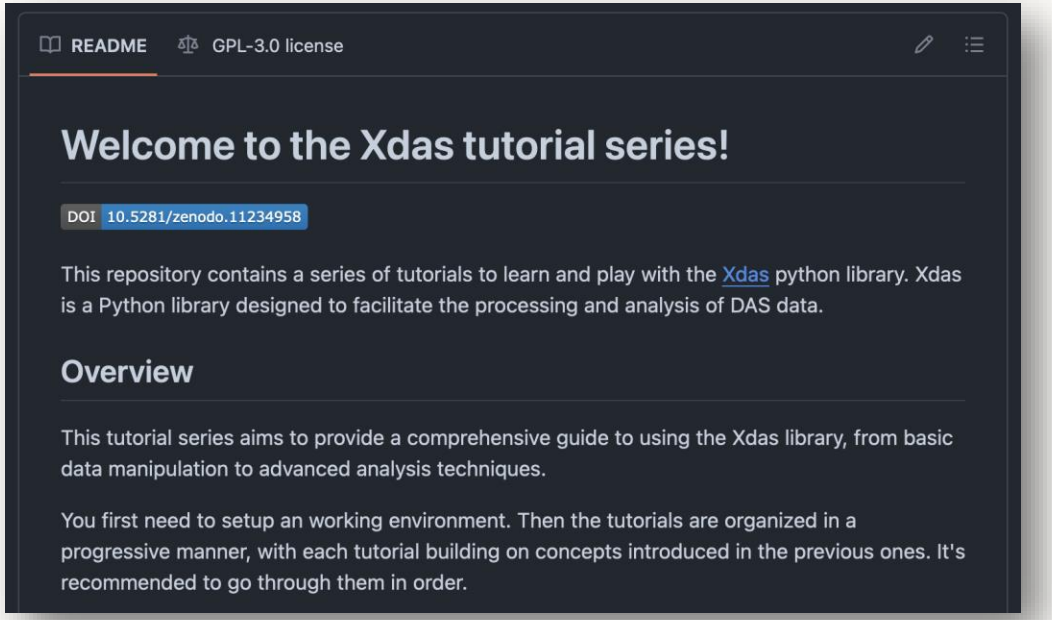

Snuffler
AB S0050 00 BN1
AB S0100 00 BN1
AB S0150 00 BN1
AB S0200 00 BN1
AB S0250 00 BN1
AB S0300 00 BN1
AB S0350 00 BN1
AB S0400 00 BN1
AB S0450 00 BN1
AB S0500 00 BN1
AB S0550 00 BN1
AB S0600 00 BN1
AB S0650 00 BN1
AB S0700 00 BN1
AB S0750 00 BN1
AB S0800 00 BN1
AB S0850 00 BN1
AB S0900 00 BN1
AB S0950 00 BN1
AB S1000 00 BN1
AB S1050 00 BN1
AB S1100 00 BN1
AB S1150 00 BN1
AB S1200 00 BN1
AB S1250 00 BN1
AB S1300 00 BN1
AB S1350 00 BN1
AB S1400 00 BN1
AB S1450 00 BN1
AB S1500 00 BN1
15:48:30
15:49:00
15:49:30
15:50:00
15:50:30
15:51:00
Nov 18, 2024
2024-11-18 15:48:29 .681

Resources

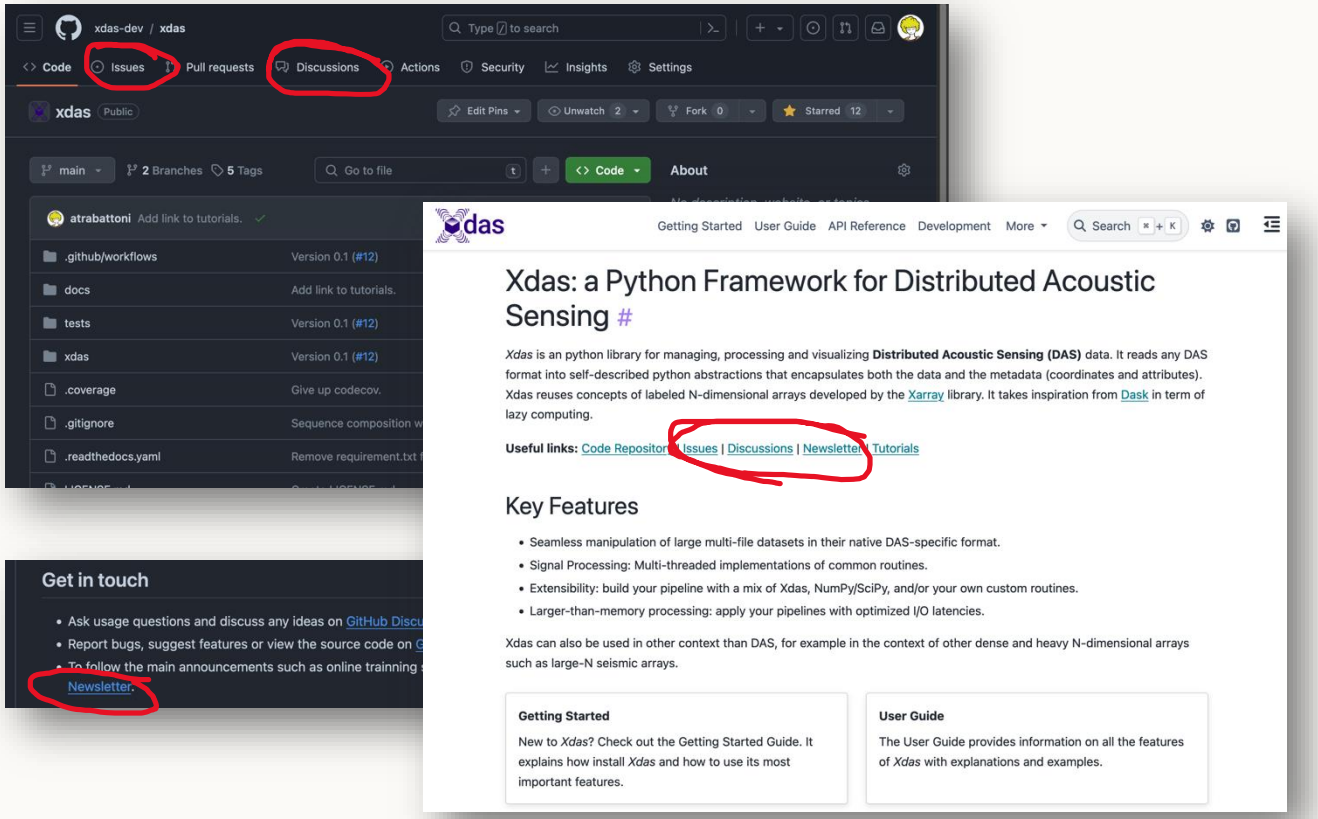
Documentation



Tutorials



Get in touch

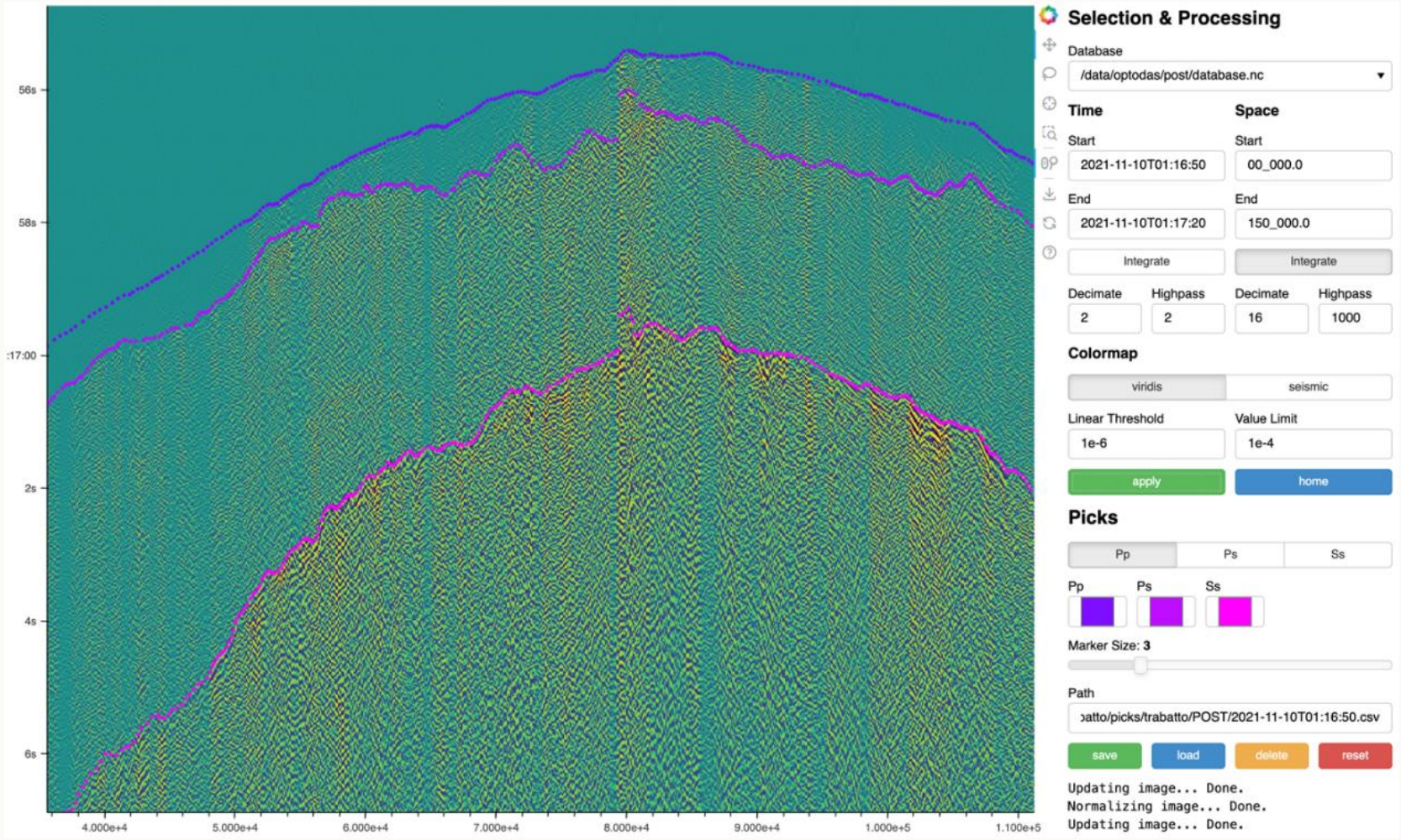


- **Newsletter**: get the main announcements.
- **Discussions**: ask questions, share ideas.
- **Issues**: Report a bug, request a feature.

Two ways to contribute

1. Join the team and improve Xdas
2. Create your toolbox within the xdas-dev github organization

Additional toolboxes: Xpick



Questions? Comments? Requests? Ideas?