Data organization

```
# TDT Tank directory: FCSV_EEG-220215-173221
# TDT Block directory: Paul-220607-102425
>> gfindlay@tononi-2 /V/f/D/F/0/T/F/Paul-220607-102425> pwd
/Volumes/fscv/Data/FSCV4-Paul/06-07-2022/TDT/FCSV_EEG-220215-173221/Paul-220607-102425

# Block directory contents
>> gfindlay@tononi-2 /V/f/D/F/0/T/F/Paul-220607-102425> ls -la
total 729608
drwxrwxr-x 2 root neuropixels         0 Jun  8 14:21 ./
drwxrwxr-x 2 root neuropixels         0 Jun  8 14:20 ../
-rw-rw-r-- 1 root neuropixels       124 Jun  7 10:24 desktop.ini
-rw-rw-r-- 1 root neuropixels  86903080 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425_EEGf_Ch1.sev
-rw-rw-r-- 1 root neuropixels  86903080 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425_EEGf_Ch2.sev
-rw-rw-r-- 1 root neuropixels  86903080 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425_EEGf_Ch3.sev
-rw-rw-r-- 1 root neuropixels  86903080 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425_EEGf_Ch4.sev
-rw-rw-r-- 1 root neuropixels  86903080 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425_EEGr_Ch1.sev
-rw-rw-r-- 1 root neuropixels  86903080 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425_EEGr_Ch2.sev
-rw-rw-r-- 1 root neuropixels  86903080 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425_EEGr_Ch3.sev
-rw-rw-r-- 1 root neuropixels  86903080 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425_EEGr_Ch4.sev
-rw-rw-r-- 1 root neuropixels   2883197 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425.Tbk
-rw-rw-r-- 1 root neuropixels  17597196 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425.Tdx
-rw-rw-r-- 1 root neuropixels       120 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425.tev
-rw-rw-r-- 1 root neuropixels     21011 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425.tin
-rw-rw-r-- 1 root neuropixels        23 Jun  7 10:24 FCSV_EEG-220215-173221_Paul-220607-102425.tnt
-rw-rw-r-- 1 root neuropixels 151821720 Jun  7 20:17 FCSV_EEG-220215-173221_Paul-220607-102425.tsq
-rw-rw-r-- 1 root neuropixels       376 Jun  7 10:24 StoresListing.txt
```

In [ ]:
```
import neo
from pathlib import Path

block_path = Path(f"/Volumes/fscv/Data/FSCV4-Paul/06-07-2022/TDT/FCSV_EEG-220215-173221/")
nio = neo.io.TdtIO(block_path)
nio
```

Out[ ]:
```
TdtIO: /Volumes/fscv/Data/FSCV4-Paul/06-07-2022/TDT/FCSV_EEG-220215-173221
nb_block: 1
nb_segment:  [1]
signal_streams: [b'EEGf' (chans: 4), b'EEGr' (chans: 4)]
signal_channels: [b'EEGf' 1, b'EEGf' 2, b'EEGf' 3, b'EEGf' 4]
spike_channels: []
event_channels: [PC1/, PC2/]
```

In [ ]:
```
assert nio.block_count() == 1
blk = nio.read_block()
blk
```

Out[ ]:
```
Block with 1 segments, 2 groups
file_origin: '/Volumes/fscv/Data/FSCV4-Paul/06-07-2022/TDT/FCSV_EEG-220215-173221'
# segments (N=1)
0: Segment with 2 analogsignals, 2 events
   # analogsignals (N=2)
   0: AnalogSignal with 4 channels of length 0; units V; datatype float32
      name: "b'EEGf'"
      annotations: {'stream_id': '0'}
      sampling rate: 1.0 Hz
      time: 0.0 s to 0.0 s
   1: AnalogSignal with 4 channels of length 0; units V; datatype float32
      name: "b'EEGr'"
      annotations: {'stream_id': '1'}
      sampling rate: 1.0 Hz
      time: 0.0 s to 0.0 s
```

In [ ]:
```
assert nio.segment_count(0) == 1
seg = nio.read_segment(lazy=False)
seg
```

Out[ ]:
```
Segment with 2 analogsignals, 2 events
# analogsignals (N=2)
0: AnalogSignal with 4 channels of length 0; units V; datatype float32
   name: "b'EEGf'"
   annotations: {'stream_id': '0'}
   sampling rate: 1.0 Hz
   time: 0.0 s to 0.0 s
1: AnalogSignal with 4 channels of length 0; units V; datatype float32
   name: "b'EEGr'"
   annotations: {'stream_id': '1'}
   sampling rate: 1.0 Hz
   time: 0.0 s to 0.0 s
```

In [ ]:
```
seg.analogsignals[1]
```

Out[ ]:
```
AnalogSignal with 4 channels of length 0; units V; datatype float32
name: "b'EEGr'"
annotations: {'stream_id': '1'}
sampling rate: 1.0 Hz
time: 0.0 s to 0.0 s
```

In [ ]:
```
lazy_seg = nio.read_segment(lazy=True)
lazy_seg
```

Out[ ]:
```
Segment with 2 analogsignals, 2 events
# analogsignals (N=2)
0: AnalogSignalProxy name: "b'EEGf'" annotations: {'stream_id': '0'}
1: AnalogSignalProxy name: "b'EEGr'" annotations: {'stream_id': '1'}
```

In [ ]:
```
lazy_seg.analogsignals[1].load()
```

Out[ ]:
```
AnalogSignal with 4 channels of length 0; units V; datatype float32
name: "b'EEGr'"
annotations: {'stream_id': '1'}
sampling rate: 1.0 Hz
time: 0.0 s to 0.0 s
```

In [ ]:
```
nio._sigs_lengths
```

Out[ ]:
```
{0: {0: 0, 1: 0}}
```

In [ ]:
```
tbk = neo.rawio.tdtrawio.read_tbk("/Volumes/fscv/Data/FSCV4-Paul/06-07-2022/TDT/FCSV_EEG-220215-173221/Paul-220607-102425/FCSV_EEG
tbk
```

Out[ ]:
```
array([(b'PC1/', b'PC1//',  True, 3, 0, 1,   257,  0, 4,   0.      ),
       (b'PC1\\', b'PC1/\\',  True, 3, 0, 2,   258,  0, 4,   0.      ),
       (b'PC2/', b'PC2//',  True, 3, 0, 1,   257,  0, 4,   0.      ),
       (b'PC2\\', b'PC2/\\',  True, 3, 0, 2,   258,  0, 4,   0.      ),
       (b'EEGf', b'EEGf',  True, 5, 4, 0, 33025, 64, 0, 610.351563),
       (b'EEGr', b'EEGr',  True, 5, 4, 0, 33025, 64, 0, 610.351563)],
      dtype=[('StoreName', 'S4'), ('HeadName', 'S16'), ('Enabled', '?'), ('CircType', '<i8'), ('NumChan', '<i8'), ('StrobeMode',
'<i8'), ('TankEvType', '<i4'), ('NumPoints', '<i8'), ('DataFormat', '<i8'), ('SampleFreq', '<f8')])
```

In [ ]:
```
nio._tsq
```

Out[ ]:
```
[array([(151821720,     0, b'',     0,     0, 0.00000000e+00,                  0, 0,   0.     ),
        (       20, 34817, b'\x01',     0,     0, 1.65461547e+09,                 40, 0,   0.     ),
        (       74, 33041, b'EEGr',     4,     0, 1.65461547e+09,                 40, 0, 610.35156),
        ...,
        (       10,   257, b'PC1/',     0,     0, 1.65465106e+09, 4692833644459327488, 4,   0.     ),
        (       10,   258, b'PC1\\', 17232, 12081, 1.65465106e+09, 4692833644459327488, 4,   0.     ),
        (       20, 34817, b'\x02',     0,     0, 1.65465106e+09,                 80, 0,   0.     )],
       dtype=[('size', '<i4'), ('evtype', '<i4'), ('evname', 'S4'), ('channel', '<u2'), ('sortcode', '<u2'), ('timestamp', '<f8'),
('offset', '<i8'), ('dataformat', '<i4'), ('frequency', '<f4')])]
```

In [ ]:
```
```