Project presentation

Pierre Bouvet

February 20, 2025

1 Introduction

This document is made as to present the vision of the HDF_BLS library and the software approach for the unification of all BLS data. This file is meant to be evolutive with the different needs encountered by researchers. It however presents succintly the current vision of the project and allows to better understand its development and its parts.

2 HDF5 file structure

2.1 Basic file structure

The vision of the project is to unify all BLS data into a single HDF5 file. This file will be used to store all the data and metadata of the BLS. The data will be stored in a hierarchical structure, all the data related to BLS being stored the "Data" group. This choice allows results from other techinques to be stored in the same file, only in other, independent groups, to minimize the risk of nomenclature competition between techinques. The "Data" group will contain all the data. These data are arranged in groups, whose identifiers are "Data_i". Each group can only contain one measure, in the form of an array, whose dimension is however not restricted. This measure will be called "Raw_data". The attributes of the measure are stored in the group attributes.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group)
|   +-- Data_1 (group)
|   |   +-- Raw_data (dataset)
```

2.2 Attributes

The attributes will follow a hierarchical structure. The attributes that apply to all "Data_i" will be stored in the "Data" group, while the parameters that apply

to a specific "Data_i" will be stored in the "Data_i" group. All the attributes are stored as text. Attributes are then divided in four categories:

- Attributes that are specific to the spectrometer used, such as the wavelength of the laser, the type of laser, the type of detector, etc. These attributes are recognized by the capital letter word "SPECTROMETER" in the name of the attribute.
- Attributes that are specific to the sample, such as the date of the measurement, the name of the sample, etc. These attributes are recognized by the capital letter word "MEASURE" in the name of the attribute.
- Attributes that are specific to the original file format, such as the name of the file, the date of the file, the version of the file, the precision used on the storage of the data, etc. These attributes are recognized by the capital letter word "FILEPROP" in the name of the attribute.
- Attributes that are used inside the HDF5 file, such as the name of the group, the name of the dataset, etc. These attributes are the only ones without a prefix.

The name of the attribues contains the unit of the attribute if it has units, in the shape of an underscore followed by the unit in parenthesis. Some parameters that can be represented by a series of norms will also be defined in a given norm, such as the ISO8601 for the date. These norms are however not specified in the name of the attribute. Here are some examples of attributes:

- "SPECTROMETER.Detector_Type" is the type of the detector used.
- "MEASURE.Sample" is the name of the sample.
- "MEASURE.Exposure_(s)" is the exposire of the sample given in seconds
- "MEASURE.Date_of_measurement" is the date of the measurement.
- "FILEPROP.Name" is the name of the file.

To unify the name of attributes, a spreadsheet is accesible, containing all the attributes and their units. This spreadsheet will be updated as new attributes are added to the project and defined with a version number that will also be stored in the attribtutes of each data attributes (under FILEPROP.version). This spreadsheet is meant to be exported in a CSV file that can be used to update the attributes of the data.

2.3 Meta-files

It can be useful to store in a same file, measures coming from different instruments, taken in different conditions, or that we just want to separate from other groups of measures. In that end, we propose a tree-like structure of the HDF5 file, where each group can contain sub-groups, which can also contain sub-groups

etc. In order to unify the way we access these groups, we propose to identify them by a unique identifier of the form "Data_i", where "i" is an integer. Here is an example of the structure of a meta-file:

```
file.h5
+-- Data
|   +-- Data_1
|   |   +-- Data_1
|   |   |   +-- ...
|   |   +-- Data_2
|   |   |   +-- ...
|   |   +-- ...
|   +-- Data_2
|   |   +-- Data_1
|   |   |   +-- Data_1
|   |   |   |   +-- ...
|   |   |   +-- Data_2
|   |   |   |   +-- ...
|   |   |   ...
|   |   ...
|   ...
```

2.4 Complete structure definition

The file is intended to be used to store not only raw data but also treated data together with the parameters used for treatment. As such, we propose to complete the structure defined above with the following structure:

```
file.h5
+-- Data (group)
|   +-- Data_0 (group)
|   |   +-- Raw_data (dataset)
|   |   +-- Abscissa_0 (dataset)
|   |   +-- Abscissa_1 (dataset)
|   |   +-- ...
|   |   +-- PSD (dataset)
|   |   +-- Frequency (dataset)
|   |   +-- Treat_0(group)
|   |   |   +-- Shift (dataset)
|   |   |   +-- Shift_std (dataset)
|   |   |   +-- Linewidth (dataset)
|   |   |   +-- Linewidth_std (dataset)
|   |   +-- Treat_1(group)
|   |   ...
|   +-- Data_1 (group)
```

```
|    ...
```

The nomenclature is defined as follows:

- **"Data_i"** is the identifier of the group containing the i-th measure.
- **"Raw_data"** is the identifier of the dataset containing the raw data of the measure stored in the "Data_i" group.
- **"Abscissa_i"** is the identifier of the dataset containing the i-th abscissa array of "Raw_data" and "PSD". The dimension of this dataset is not forced to one.
- **"PSD"** is the identifier of the dataset of the Power Spectrum Density of the measure, associated with the "Frequency" dataset. "Raw_data" and "PSD" are arrays of same shape. We impose the last dimension(s) of "PSD" to be the same as the dimension(s) of "Frequency".
- **"Frequency"** is the identifier of the dataset containing the frequency axis of the "PSD".
- **"Treat_i"** is the identifier of the group containing the treated data of the i-th measure.
- **"Shift"** is the identifier of the dataset containing the values of the fitted frequency shifts.
- **"Shift_std"** is the identifier of the dataset containing the standard deviation of the fitted frequency shifts.
- **"Linewidth"** is the identifier of the dataset containing the values of the fitted linewidths.
- **"Linewidth_std"** is the identifier of the dataset containing the standard deviation of the fitted linewidths.

2.5 Examples

2.5.1 Example 1 - A single measure with no treatment

In this first example, we want to store a single measure of a water sample.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Data_0 (group) -> Name = "Water"
|   |   +-- Raw_data (dataset)
```

Note that we have here added arrows and an example of the value of the "Name" attributes.

2.5.2 Example 2 - A series of measures with no treatment

In this second example, we want to store a series of measures taken on three different samples: Water, Ethanol and Glycerol.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Data_0 (group) -> Name = "Water"
|   |   +-- Raw_data (dataset)
|   +-- Data_1 (group) -> Name = "Ethanol"
|   |   +-- Raw_data (dataset)
|   +-- Data_2 (group) -> Name = "Glycerol"
|   |   +-- Raw_data (dataset)
```

Note that we have here added arrows and an example of the value of the "Name" attributes.

2.5.3 Example 3 - A series of series of measures with no treatment but with a calibration spectrum and an impulse response measure

In this third example, we want to store a series of two measures taken on two different samples: Water and Ethanol. We also want to store a calibration curve and an impulse response curve.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Data_0 (group) -> Name = "Impulse_Response"
|   |   +-- Raw_data (dataset)
|   +-- Data_1 (group) -> Name = "Calibration"
|   |   +-- Raw_data (dataset)
|   +-- Data_2 (group) -> Name = "Water"
|   |   +-- Data_0 (group) -> Name = "Water_01"
|   |   |   +-- Raw_data (dataset)
|   |   +-- Data_1 (group) -> Name = "Water_02"
|   |   |   +-- Raw_data (dataset)
|   +-- Data_3 (group) -> Name = "Ethanol"
|   |   +-- Data_0 (group) -> Name = "Ethanol_01"
|   |   |   +-- Raw_data (dataset)
|   |   +-- Data_1 (group) -> Name = "Ethanol_02"
|   |   |   +-- Raw_data (dataset)
```

Note that we have here added arrows and an example of the value of the "Name" attributes.

2.5.4 Example 4 - A single measure converted to a Power Spectrum Density

In this fourth example, we want to store a single measure of a water sample. This measure has been converted into a Power Spectrum Density.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Data_0 (group) -> Name = "Water"
|   |   +-- Raw_data (dataset)
|   |   +-- PSD (dataset)
|   |   +-- Frequency (dataset)
```

Note that we have here added arrows and an example of the value of the "Name" attributes. In this case, all the steps of the conversion to PSD are stored in the "Data_0" group. The nomenclature of the attribute(s) used to store the parameters of the treatment is not specified.

2.5.5 Example 5 - Multiple measures converted to a Power Spectrum Density with a time-independent spectrometer

In this fifth example, we are in the situation where a time-independent spectrometer has been used to acquire multiple measures. In this case, the hierrarchy of the file can be used to reduce the number of datasets, by considering that all the PSD share the same frequency axis.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Frequency (dataset)
|   +-- Data_0 (group) -> Name = "Sample_1"
|   |   +-- Raw_data (dataset)
|   |   +-- PSD (dataset)
|   +-- Data_1 (group) -> Name = "Sample_2"
|   |   +-- Raw_data (dataset)
|   |   +-- PSD (dataset)
```

Note that we have here added arrows and an example of the value of the "Name" attributes.

2.5.6 Example 6 - A single measure with a treatment

In this sixth example, we want to store a single measure of a water sample that has been treated.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Data_0 (group) -> Name = "Water"
|   |   +-- Raw_data (dataset)
|   |   +-- PSD (dataset)
|   |   +-- Frequency (dataset)
|   |   +-- Treat_0 (group) -> Name = "Treat_5GHz"
|   |   |   +-- Shift (dataset)
|   |   |   +-- Shift_std (dataset)
|   |   |   +-- Linewidth (dataset)
|   |   |   +-- Linewidth_std (dataset)
```

Note that we have here added arrows and an example of the value of the "Name" attributes. In this case, all the steps of the treatment are stored in the "Treat_0" group. The nomenclature of the attribute(s) used to store the parameters of the treatment is not specified.

2.5.7 Example 7 - A single measure with two distinct treatments

In this seventh example, we will store a single measure where two different treatments have been performed (for example a measure at an interface between two materials).

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Data_0 (group) -> Name = "Water"
|   |   +-- Raw_data (dataset)
|   |   +-- PSD (dataset)
|   |   +-- Frequency (dataset)
|   |   +-- Treat_0 (group) -> Name = "Treat_5GHz"
|   |   |   +-- Shift (dataset)
|   |   |   +-- Shift_std (dataset)
|   |   |   +-- Linewidth (dataset)
|   |   |   +-- Linewidth_std (dataset)
|   |   +-- Treat_1 (group) -> Name = "Treat_10GHz"
|   |   |   +-- Shift (dataset)
|   |   |   +-- Shift_std (dataset)
|   |   |   +-- Linewidth (dataset)
|   |   |   +-- Linewidth_std (dataset)
```

Note that we have here added arrows and an example of the value of the "Name" attributes. In this case, all the steps of the treatment around 5GHz are stored in the "Treat_0" group and the ones around 10GHz in the "Treat_1" group. The nomenclature of the attribute(s) used to store the parameters of the treatment is not specified.

2.5.8 Example 8 - A single mapping stored as a single measure

In this eighth example, we want to store a mapping of a sample. This mapping has been obtained with a spectrometer that returns an array of points for all the points mapped. To clarify this example, we will indicate the dimension of each dataset here between brackets.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Data_0 (group) -> Name = "Sample"
|   |   +-- Raw_data (dataset) [X, Y, M]
|   |   +-- PSD (dataset) [X, Y, N]
|   |   +-- Frequency (dataset) [N]
|   |   +-- Abscissa_0 (dataset) [X] -> Name = "x (mm)"
|   |   +-- Abscissa_1 (dataset) [Y] -> Name = "y (mm)"
|   |   +-- Treat_1 (group) -> Name = "Treat"
|   |   |   +-- Shift (dataset) [X, Y]
|   |   |   +-- Shift_std (dataset) [X, Y]
|   |   |   +-- Linewidth (dataset) [X, Y]
|   |   |   +-- Linewidth_std (dataset) [X, Y]
```

Note that we have here added arrows and an example of the value of the "Name" attributes.

2.5.9 Example 9 - A series of mapping over the same field of view stored as a single measure

In this ninth example, we are in the situation where multiple mappings of same dimension have been obtained with a spectrometer that returns an array of points for all the points mapped. In this case, the hierrarchy of the file can be used to reduce the number of datasets, by considering that all the PSD share the same frequency axis and the same field of view.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Abscissa_0 (dataset) [X] -> Name = "x (mm)"
|   +-- Abscissa_1 (dataset) [Y] -> Name = "y (mm)"
|   +-- Frequency (dataset) [N]
|   +-- Data_0 (group) -> Name = "Day_1"
|   |   +-- Raw_data (dataset) [X, Y, M]
|   |   +-- PSD (dataset) [X, Y, N]
|   |   +-- Treat_1 (group) -> Name = "Treat"
|   |   |   +-- Shift (dataset) [X, Y]
|   |   |   +-- Shift_std (dataset) [X, Y]
|   |   |   +-- Linewidth (dataset) [X, Y]
```

```
|   |   |   +-- Linewidth_std (dataset) [X, Y]
|   +-- Data_1 (group) -> Name = "Day_2"
|   |   +-- Raw_data (dataset) [X, Y, M]
|   |   +-- PSD (dataset) [X, Y, N]
|   |   +-- Treat_1 (group) -> Name = "Treat"
|   |   |   +-- Shift (dataset) [X, Y]
|   |   |   +-- Shift_std (dataset) [X, Y]
|   |   |   +-- Linewidth (dataset) [X, Y]
|   |   |   +-- Linewidth_std (dataset) [X, Y]
```

Note that we have here added arrows and an example of the value of the "Name" attributes.

2.5.10 Example 10 - A series of mapping over the same field of view stored as multiple measures

In this tenth example, we are in the situation where multiple mappings of same dimension have been obtained with a spectrometer that can't return an array of points for all the points mapped, but returns them one by one. Because it would be impractical to create groups for each point, we encourage users to compile their data into a single dataset, and refer to example 9.

2.5.11 Example 11 - A series of mapping obtained with different spectrometers and with different field of view

In this eleventh example, we are in the situation where multiple mappings of different dimensions have been obtained with different spectrometers that all return an array of points for all the points mapped. In this case, the hierrarchy of the file cannot be used to reduce the number of datasets, and each group will need its own abscissa and frequency datasets.

The following structure represents the base structure of the file:

```
file.h5
+-- Data (group) -> Name = "Measure"
|   +-- Data_0 (group) -> Name = "VIPA"
|   |   +-- Raw_data (dataset) [X, Y, M]
|   |   +-- PSD (dataset) [X, Y, N]
|   |   +-- Abscissa_0 (dataset) [X] -> Name = "x (mm)"
|   |   +-- Abscissa_1 (dataset) [Y] -> Name = "y (mm)"
|   |   +-- Frequency (dataset) [N]
|   |   +-- Treat_1 (group) -> Name = "Treat"
|   |   |   +-- Shift (dataset) [X, Y]
|   |   |   +-- Shift_std (dataset) [X, Y]
|   |   |   +-- Linewidth (dataset) [X, Y]
|   |   |   +-- Linewidth_std (dataset) [X, Y]
|   +-- Data_1 (group) -> Name = "TFP"
```

```
|   |   +-- Raw_data (dataset) [X, Y, M]
|   |   +-- PSD (dataset) [X, Y, N]
|   |   +-- Abscissa_0 (dataset) [X] -> Name = "x (mm)"
|   |   +-- Abscissa_1 (dataset) [Y] -> Name = "y (mm)"
|   |   +-- Frequency (dataset) [N]
|   |   +-- Treat_1 (group) -> Name = "Treat"
|   |   |   +-- Shift (dataset) [X, Y]
|   |   |   +-- Shift_std (dataset) [X, Y]
|   |   |   +-- Linewidth (dataset) [X, Y]
|   |   |   +-- Linewidth_std (dataset) [X, Y]
```

Note that we have here added arrows and an example of the value of the "Name" attributes.

3 Pipelines

3.1 General view

The goal of this software is to go from a storaged data file to a usable data set with a reproducible, stable and unified treatment protocol. To schematize this treatment protocol, we propose the following diagram: