Human Intracranial Brain Observations Player User Manual

BONTEMPS Benjamin
CHATARD Benoît
GANNERIE Adrien

February 28, 2018

Abstract

This document is the user manual of the Human Intracranial Brain Observations Player (HiBoP). HiBoP is being developed by the Centre de Recherche en Neurosciences de Lyon (CRNL), with funds from the Human Brain Project (HBP) to visualize large datasets of intracranial electroencephalography (iEEG) data. This document was written for the 2.1.0 version of HiBoP.

Contents

1 HiBoP project

A **Project** links together anatomical and functional data from a set of patients. If you have an anatomical description of the patients (3D mesh of the

brain, MRI and coordinates of the sites of the electrodes) and the respective iEEG data, you can visualize them in 4D (3D space + time). The first thing to do with HiBoP is to create or open a Project. This can be done within HiBoP (section 2), but also using a third-party software which automatically generates the files and the hierarchy, such as the HiBoP Database Manager (appendix A).

On your Hard Drive, the Project consists of a folder with several subfolders (Figure 1). Each file in this subfolder is encoded with the json format, and thus can be edited (at your own risks) using any text editor. However this is not recommended, as any coma missing may cause the file to be incorrectly interpreted.

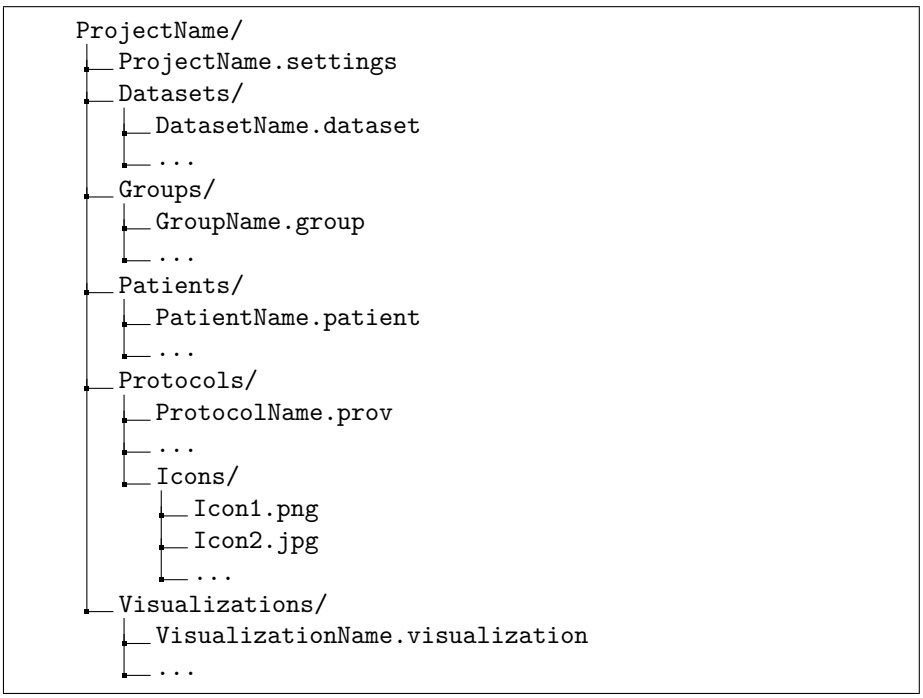

Figure 1: Project directory hierarchy

1.1 Settings

The .settings file at the root of the projet folder describes some project settings (Figure 2). This file contains a default path for the anatomical data of the patients and a default path for the functional data (iEEG).

```
{
  "ID": "41001a8a-23a5-409b-8b87-099dbb727a18",
  "Name": "ProjectName",
  "PatientDatabase": "Y:/BrainVisaDB/Epilepsy",
  "LocalizerDatabase": "Y:/LOCA_DATABASE"
}
```

Figure 2: Example project settings file

1.2 Patients

This subfolder contains .patient files, which contains metadata of the patients and full paths towards anatomical files for each patient (Figure 3). A 3D mesh can be a single mesh (only one file) or a left-right mesh (one file for the left part and one for the right part of the brain). These files are in GIfTI format (.gii). You can also provide aditional files to show MarsAtlas parcels (.gii), or a transformation file (to align the mesh and its respective MRI, .trm). The MRIs must be in the NIfTI-1 format (.nii). The implantations must be in the PTS format. The connectivities files are CCEP (amplitude / latencies) text files.

```
{
  "ID": "LYONNEURO_2013_CHAj",
  "Name": "CHAj",
  "Date": 2013,
  "Place": "LYONNEURO",
  "Brain": {
    "Meshes": [
      {
        "$type": "HBP.Data.Anatomy.LeftRightMesh, Assembly-CSharp",
        "ID": "bb1ee4e1-b96b-4973-814d-fb30c4d0d512",
        "Name": "Grey matter",
        "LeftHemisphere": "[...]/LYONNEURO_2013_CHAj_Left.gii",
        "RightHemisphere": "[...]/LYONNEURO_2013_CHAj_Right.gii",
        "LeftMarsAtlasHemisphere": "",
        "RightMarsAtlasHemisphere": "",
        "Transformation": "[...]/LYONNEURO_2013_CHAj.trm"
      }
    ],
    "MRIs": [
      {
        "Name": "Preimplantation",
        "File": "[...]/LYONNEURO_2013_CHAj.nii"
      }
    ],
    "Connectivities": [],
    "Implantations": [
      {
        "Name": "Patient",
        "File": "[...]/LYONNEURO_2013_CHAj.pts",
        "MarsAtlas": "[...]/LYONNEURO_2013_CHAj.csv"
      },
      {
        "Name": "MNI",
        "File": "[...]/LYONNEURO_2013_CHAj_MNI.pts",
        "MarsAtlas": "[...]/LYONNEURO_2013_CHAj.csv"
      }
    ],
    "Epilepsy": {
      "Type": 5
    }
  }
}
```


Figure 3: Example patient file

1.3 Groups

This subfolder contains .group files, which are simple lists of patients (Figure 4).

```
{
  "ID": "{813cbd0a-550c-40de-95c7-b387626f9a96}",
  "Name": "VISU",
  "Patients": [
    "Gre_2013_BONn",
    "Gre_2015_BARr",
    "LYONNEURO_2013_CARj",
    "LYONNEURO_2017_PECv",
    "Gre_2015_PATa",
    "LYONNEURO_2014_REId"
  ]
}
```


Figure 4: Example group file

1.4 Protocols

This subfolder contains .prov files (Figure 5). The .prov extension stands for **protocol visualization**. Typically, a functional data file in HiBoP is a set of time series (one for each channel) with data for each sample from the beginning to the end of the recording session. Such data are stored in the .eeg/.eeg.ent[1] files. In addition, information about the events of the experimental paradigm is stored in .pos files : event codes occurred at specific samples (e.g. an event of type « 10 » occurred at sample 134567). The .prov file instructs HiBoP on how to epoch the data based on the .pos file. The idea is to create blocs of epochs, each of them centered around a specific type of events, called **main events** (i.e. consider all events with code « 10 » and extract for each of them a window ranging from 200 ms before to 700 ms after that event). Prov files define the blocs you want to visualize (code of the main event), how you want to call them, how you want to sort epochs within a bloc, etc. When you define a new bloc, you must specify the main event (its code) and you can specify secondary events. Secondary events are useful when you want to sort trials: say that the task shows a picture (of type 10 = faces, 20 = landscapes, or 30 = fruits) and that participants have to press one or two buttons to specify whether the picture shows a fruit (1 = « yes », 2 = « no »). You would create for instance a bloc with all epochs around an event of type « 10 » (faces) and sort those epochs according to reaction time (the latency of event 1 or 2 relative to event 10 within each epoch). To do that in HiBoP, you would specify a secondary event in the « face » bloc with type 1 and 2 (i.e. all events of type 1 or 2), then write the « sorting rule » C0L1: which means that you want to sort trials according to the code of the main event (here 10, so no sorting necessary), then for all ex-aequos epochs, use the relative latency of the secondary event (1 or 2, relative to the main event 10). You could set the main event to 10 and 20 and 30 to include in your bloc « faces », « landscapes » and « fruits », then the « sorting rule » C0L1 would sort trials according to the main event code first (all faces, then all landscapes, then all fruits) and then, within a given category, sort according to reaction time (secondary event 1 and 2) The baseline is used for the trial matrices, which can include a baseline correction. This baseline correction is described in preferences (section 2.1). The rest of the .prov file

[1]ELAN format – ELAN is freely available at http://elan.lyon.inserm.fr

is straightforward : you can ask HiBoP to show .png or .jpg files at specific latencies relative to the main event.


```
{
  "ID": "{9fd88c3e-ce7f-4c1c-83b1-f4d1edcbfe5a}",
  "Name": "LEC1",
  "Blocs": [
    {
      "IllustrationPath": "./Protocols/Icons/LEC1_SEM.jpg",
      "ID": "d3ce596a-8918-4ba1-829d-5aa73efe060e",
      "Name": "SEMAN",
      "Position": {
        "Row": 1,
        "Column": 1
      },
      "Sort": "COL1",
      "Window": {
        "Start": -500.0,
        "End": 3500.0
      },
      "Baseline": {
        "Start": -200.0,
        "End": 0.0
      },
      "Events": [
        {
          "Name": "SEM",
          "Codes": [
            10
          ],
          "Type": 0
        },
        {
          "Name": "RESPONSE",
          "Codes": [
            1,
            2
          ],
          "Type": 1
        }
      ],
      "Scenario": {
        "Icons": [
          {
            "IllustrationPath": "./Protocols/Icons/LEC1_SEM.jpg",
            "Name": "PIC",
            "Window": {
              "Start": 0.0,
              "End": 2000.0
            }
          }
        ]
      }
    }
  ]
}
```


Figure 5: Example prov file (one bloc of the LEC1 LOCA)

1.5 Datasets

This subfolder contains .dataset files, which summarize and link together anatomical and functional data for a given cognitive task (Figure 6). A dataset is a list of data information, each of them connecting one patient (and its anatomical data) with functional data (.eeg and .pos files) for a single experiment. Each data information can be normalized indepently from what is set in the preferences (see Section 2.1 for detailed information about normalization).


```
{
  "ID": "{dfad2720-a0ec-49d8-b552-787f28c78c3b}",
  "Name": "VISU_dataset",
  "Protocol": "{9d12c695-7ce6-49d4-82d0-a9da12ba776a}",
  "Data": [
    {
      "Name": "alpha_beta",
      "Patient": "Gre_2013_BONn",
      "Measure": "EEG data",
      "EEG": "[...]/Gre_2013_BONn_VISU_f8f24_ds8_sm0.eeg",
      "POS": "[...]/Gre_2013_BONn_VISU_ds8.pos",
      "Normalization": 4
    },
    {
      "Name": "alpha_beta",
      "Patient": "Gre_2015_BARr",
      "Measure": "EEG data",
      "EEG": "[...]/GRE_2015_BARr_VISU_f8f24_ds16_sm0.eeg",
      "POS": "[...]/GRE_2015_BARr_VISU_ds16.pos",
      "Normalization": 4
    },
    {
      "Name": "alpha_beta",
      "Patient": "Gre_2014_MADl",
      "Measure": "EEG data",
      "EEG": "[...]/GRE_2014_MADl_VISU_f8f24_ds8_sm0.eeg",
      "POS": "[...]/GRE_2014_MADl_VISU_ds8.pos",
      "Normalization": 4
    }
  ]
}
```


Figure 6: Example dataset file

1.6 Visualizations

This subfolder contains .visualization files (Figure 7), which describe the contents of a visualization: it is composed of patients and columns. Each column allow one to visualize either the anatomy of the chosen patients or iEEG data for a specific protocol/bloc/dataset/data. The also stores the configuration of the visualization or of a particular column or site.

```
{
  "ID": "25c11585-fb23-4c70-b4d3-504f9851da11",
  "Name": "Visualization",
  "Patients": [
    "Gre_2009_CHIm",
    "Gre_2009_MWAm",
    "LYONNEURO_2013_CARj",
    "LYONNEURO_2013_DUMp"
  ],
  "Configuration": {
    "Brain Color": 15,
    "Brain Cut Color": 14,
    "Colormap": 13,
    "Mesh Part": 2,
    "Mesh": "MNI",
    "MRI": "MNI",
    "Implantation": "MNI",
    "Edges": false,
    "Sites": false,
    "MRI Min": 0.0,
    "MRI Max": 1.0,
    "Camera Type": 0,
    "Cuts": [],
    "Views": []
  },
  "Columns": [
    {
      "Dataset": "{dfad2720-a0ec-49d8-b552-787f28c78c3b}",
      "Protocol": "{9d12c695-7ce6-49d4-82d0-a9da12ba776a}",
      "Bloc": "3916b0a0-7ebc-4934-94e5-2308bf511f7c",
      "Label": "alpha_beta",
      "Configuration": {
        "ConfigurationBySite": {},
        "Site Gain": 1.0,
        "Site Maximum Influence": 15.0,
        "Transparency": 0.8,
        "Span Min": 0.0,
        "Middle": 0.0,
        "Span Max": 0.0,
        "Regions Of Interest": []
      }
    }
  ]
}
```


Figure 7: Example visualization file

2 HiBoP description

This section will describe in details how to use HiBoP to create a HiBoP project and how to visualize the iEEG data. To open HiBoP, double click on HiBoP.exe (for Windows; for Linux and MacOS, read the README.md file located next to the executable). If you have any doubt about what an interface element does, you can hover it for a second and a tooltip will appear telling you what it does.

2.1 Preferences

A lot of global parameters shared through all projects can be set using the preference menu (Figure 8). These parameters are stored in a file located next to the HiBoP executable (GeneralSettings.txt).

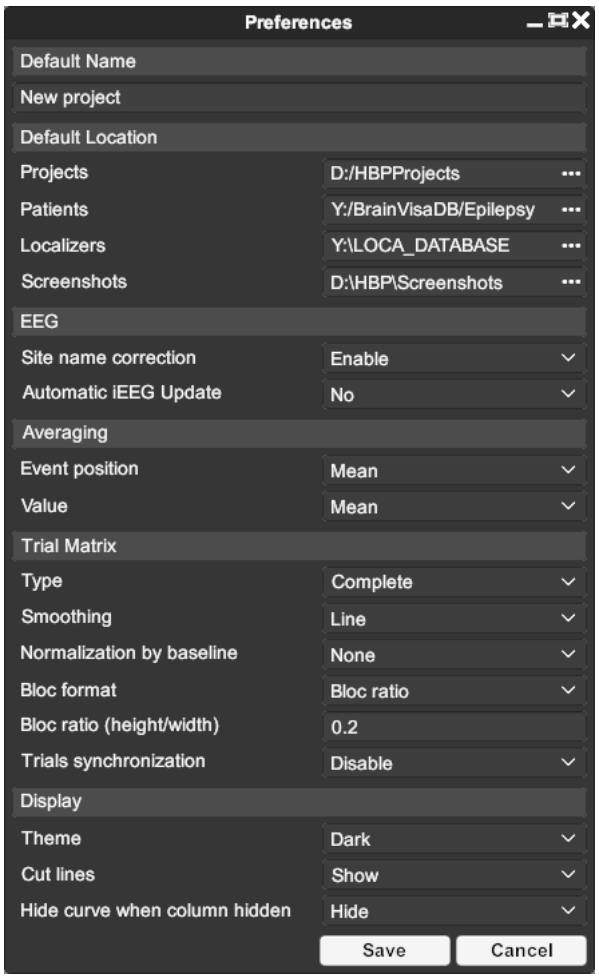

Figure 8: Preferences

- **Default Name** : Default name for a new project
- **Projects** : Default path to search for projects
- **Patients** : Default path to patients database
- **Localizers** : Default path to functional data
- **Screenshots** : Default path to save screenshots
- **Site name correction** : Change the name of a certain patern of site names (for instance "xp4" becomes "X'4")

- **Automatic iEEG Update** : Update the iEEG values on the brain automatically if they are outdated
- **Event position** : How to average the secondary events position
- **Value** : How to average the values
- **Type** : Show every matrices of a selected protocol or only blocs chosen in columns
- **Smoothing** : Should the matrices be smoothed horizontally (to have a better resolution)
- **Normalization by baseline** : The following formula is applied to each value v

  $$v = \frac{v - \mu_b}{\sigma_b}$$

  where b is an array of the baseline values depending on this parameter, μ_b is their mean and σ_b is their standard deviation
- **Bloc format** : How the trial matrices should be displayed (parameters depend on this parameter, try yourself to see which one is better for you)
- **Trial synchronization** : Should the matrices be synchronized when selecting specific trials
- **Theme** : Theme of the software (light theme is still under development)
- **Cut lines** : Should the cut lines appear on the cuts ?
- **Hide curve when column hidden** : Should the curves of the hidden column appear ?

2.2 Project management

2.2.1 Creating, opening and saving a project

When creating a new project, you have to specify the name of your new project, the location where it will be saved, the location of the anatomical database and the location of the functional database (Figure 9). The two last fields are optional.

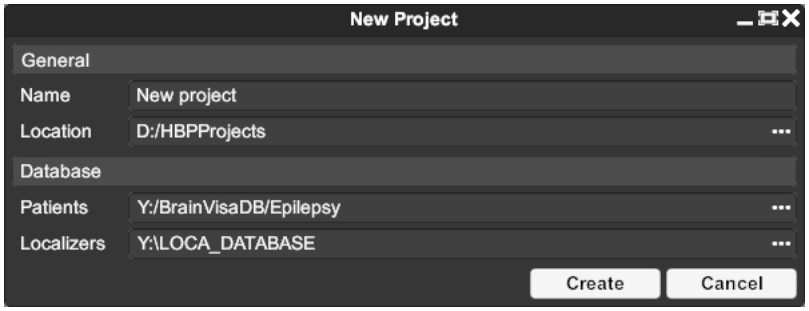

Figure 9: Create a new project

In order to open a previously created project (Figure 10), you have to tell where your projects are located. Once these are loaded, you can either double click on the project you want to open or select the project and then click on Open. You can also sort projects using specific parameters.

Figure 10: Open a project

If you want to save a project to a different location or with a different name than when it was open, you can use the "Save as" menu (Figure 11).

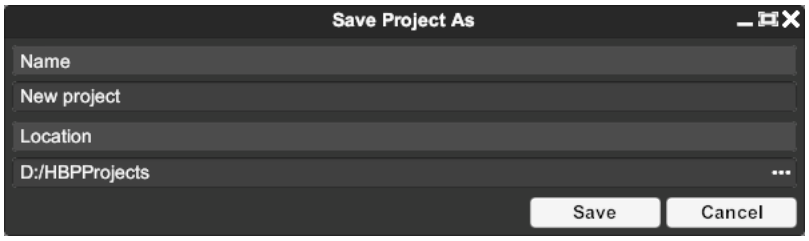

Figure 11: Save a project as

All of these menus are located in the File menu at the top of the screen.

2.2.2 Patients management

To add patients to the opened project, you have to use the patients manager (Figure 12). The window has two panels: the left panel displays all available patients in the database (found in the folder you specify at the top of the window) whereas the right panel displays patients added to this project. For each patient, numbers indicate the status of their respective anatomical data (Meshes, MRIs, Implantations and Connectivities). To add or remove patients, you can either select them in any list and use the arrows to add or remove them, or you can create new patients from scratch using the "+" button in the center. In order for the anatomical database to be correctly loaded, you have to use a specific hierarchy described in appendix B.

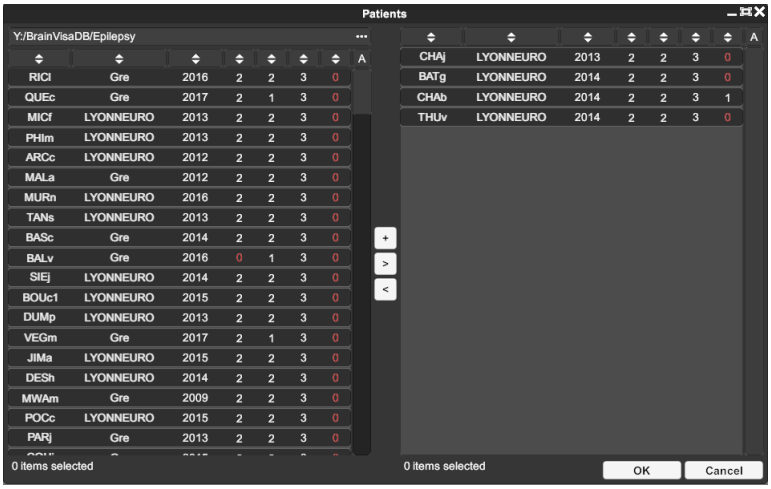

Figure 12: Patients manager

To edit any loaded patient, you can double click[2] on it. This will open the patient modifier window (Figure 13) which will allow you to edit anything concerning the metadata or the anatomical data of this patient. You can navigate between anatomical data types using the tabs in the middle of the window.

[2]The process of double clicking an item to edit it is common to most windows of HiBoP

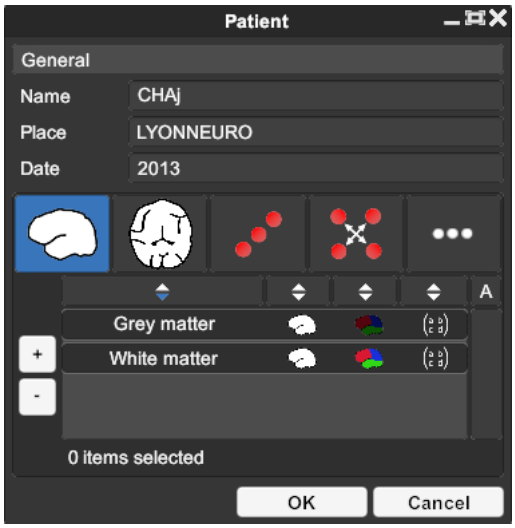

Figure 13: Patient modifier

If you want to add a mesh, you can use the "+" button located to the left; if you want to edit a mesh, you can double click the mesh item, and this will open a mesh modifier window (Figure 14) which will allow you to edit the path to the GIfTI files or to the transformation file.

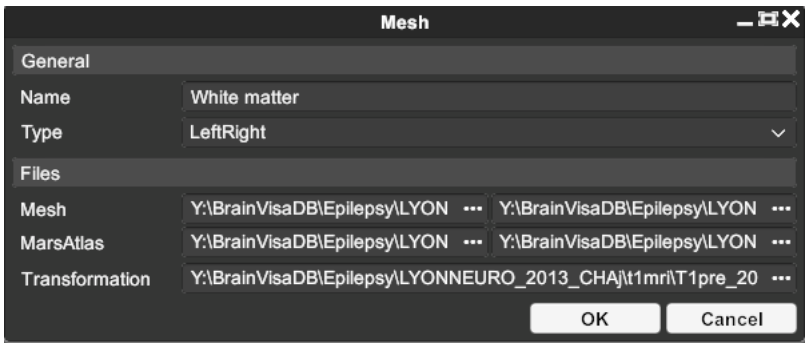

Figure 14: Mesh modifier

As stated in section 1, you can also create groups of patients to add them more easily to the visualizations. This is done through the groups manager window (Figure 15). To create a new group, use the "+" button on the left. To remove existing groups, select the group(s) you want to remove, and use the "-" button on the left[3]. Adding and removing patients from a group is very similar to the patients manager window (Figure 16).

[3]The process of using "+" button to add and "-" button to remove is common to most windows of HiBoP

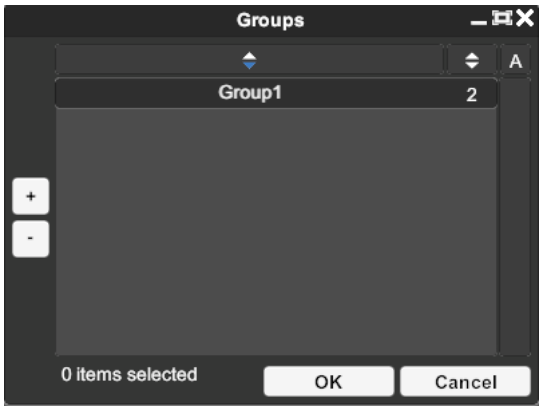

Figure 15: Groups manager

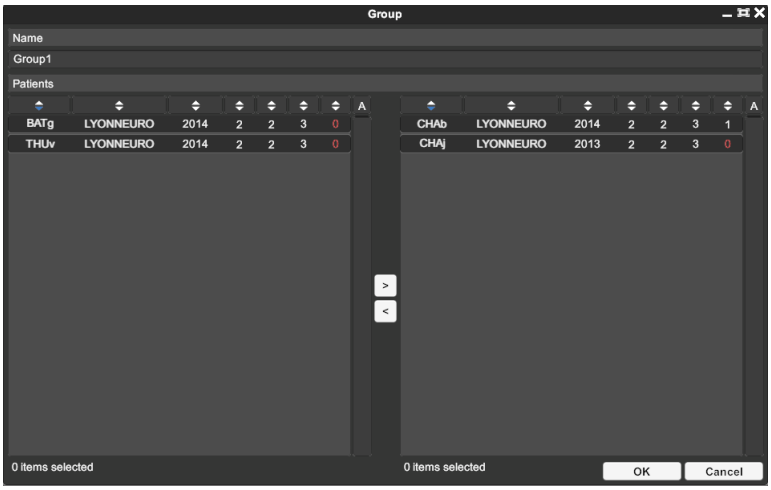

Figure 16: Group modifier

2.2.3 Data management

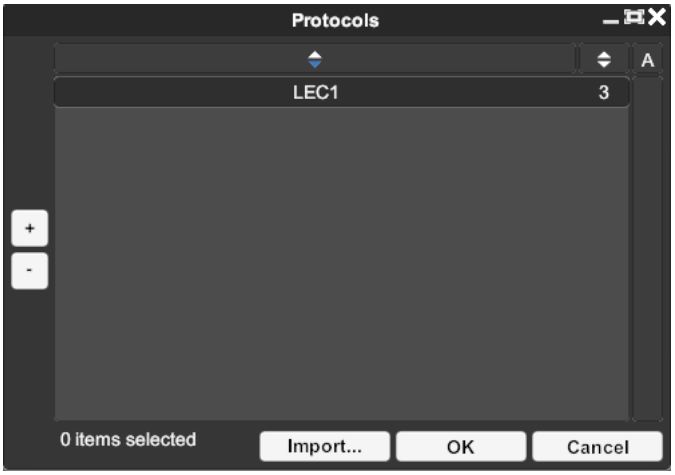

Figure 17: Protocols manager

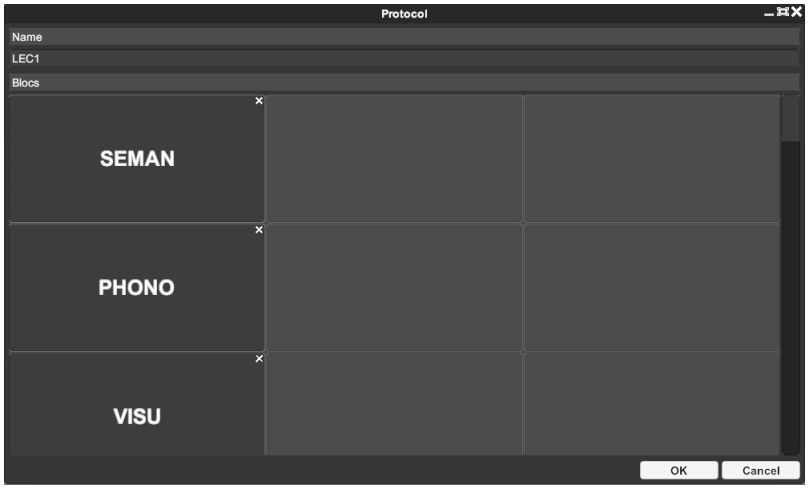

Figure 18: Protocol modifier

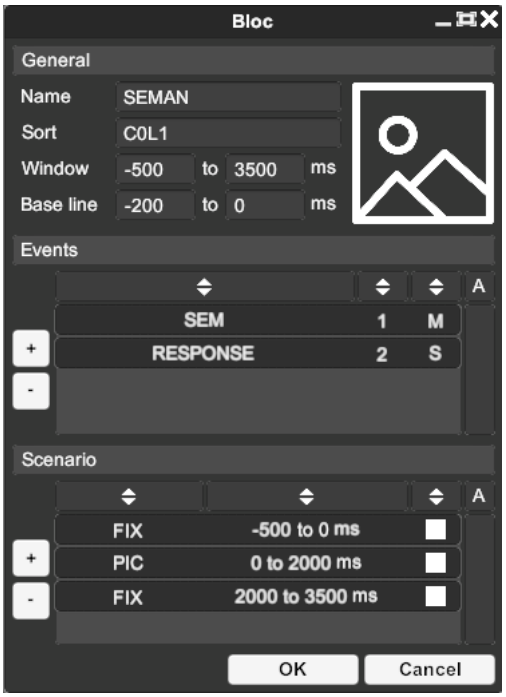

Figure 19: Bloc modifier

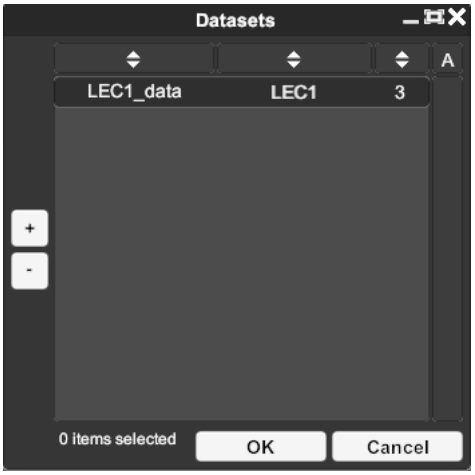

Figure 20: Datasets manager

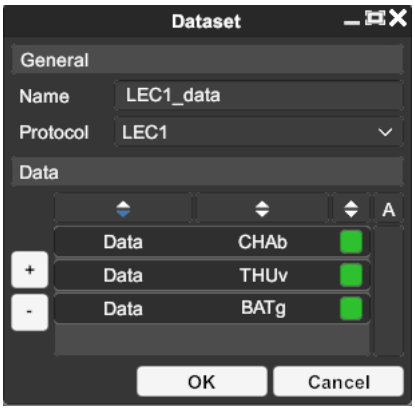

Figure 21: Dataset modifier

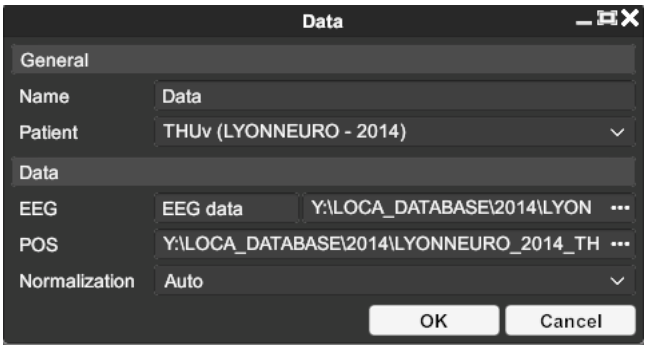

Figure 22: Data information modifier

2.2.4 Visualizations management

A HiBoP Database Manager

B HiBoP Database Hierarchy