Brenda Farrell Updated 12/14/2017 11:14:23 AM

The main script HDF52MAT_script.m creates two structures hdf52mat_out, array_of_hdf52mat_out. The first structure hdf52mat_out will be used to grab 79 fields of each specimen file and the second array_of_hdf52mat_out is the final collection of all the specimen data (from 1 to 89 specimen files) that is translated from HDF5 to MATLAB.

N is the final number of specimen files to be translated (currently from 1 (one) to 89) where beg is marker of the initial specimen file to be translated. For example if you want to only translate specimen number one (1) you will have N=1 and beg =0. If you want to translate specimen number 23 you will have N=1 and beg =22. If you want to translate specimen 23, 24, and 25 you will have N=3 and beg =22.

[array_of_hdf52mat_out, INFO]= HDF52MAT_v2(hdf52mat_out,array_of_hdf52mat_out, N, beg); .

This function returns array_of_hdf52mat_out with the requested specimen files translated and saved to an array of struct each with 79 fields. It also returns INFO a description of the last specimen file that is opened and examined with MATLAB function:

INFO = h5info(filename) returns information about the entire HDF5 file specified by filename.

The main file then translates all the data that was harvested in the array_of_hdf52mat_out to a table for visualization and for perusal. This Table does not show units, as it was not harvested but this will be implemented later.

HDF52MAT_v2 is the main function of the program it grabs the four main attributes common to each data set; the date of the recording; the original cell number, the name of the researcher who collected the data and the name of the person responsible for the data. It then grabs all data from each dataset. This is straight-forward as the information about each HDF5 file can be retrieved from INFO.

The group where the datasets are found is first established and the number of datasets within this group are found and then the properties of each dataset. For example, the Name of k dataset is found with:

```
name_d=INFO.Groups(1).Groups(1).Datasets(k).Name
```

The Class of the Datatype be it double, integer or string is found with

```
class_d=INFO.Groups(1).Groups(1).Datasets(k).Datatype.Class;
```

The Size of the Dataspace is found with

```
space_d=INFO.Groups(1).Groups(1).Datasets(k).Dataspace.Size;
```

It has one local function

[hdf2mat_out]=translate_HDF5double_or_int_2_MLB(namenew,name_g,name_d,space_d,TF,TF3,TF2,hdf2mat_out,class_d);

It sends the filename (namenew), group name (name_g), descriptions of space (space_d), type of data (class_d) found within dataset to the function. It also sends the results of three logical descriptors (TF, TF2 and TF3). If TF = 0 or TF=1 then the dataset exists or does not; if TF3 = 0 or TF3=1 then the space of dataset is either not empty or it is empty; if TF2 =0 or TF2 =1 then the name of the dataset is compatible with MATLAB rules or it is not. It returns the structure hdf52mat_out with the appropriate fields added. We use a local function here because we are reading from a file (the h5read function) and this is slow, and hence we avoid the need to change directories as data files and M files are stored in different directories. Once all the data is translated the file adds the working specimen with all fields to the array_of_hdf52mat_out and the next specimen is translated.