

---

# **LBWG memo 14**

## **Description of loop 2 HDF5 routines**

Sean Mooney, 2018.10.08

---

# 1 Introduction

## 1.1 Overview of the routines

HDF5 (Hierarchical Data Format 5) is a file format designed to store and organize large amounts of data. There is no limit on the number or size of data objects that can be stored in HDF5 files, and manipulating HDF5 files is fast. For these reasons, the HDF5 file format was chosen to handle the phase solutions in the long-baseline pipeline.

The long-baseline pipeline is structured into loops, as outlined in memo 11. Much of the book-keeping of the pipeline is done in loop 2, as it organizes data between the higher-level loop 1, which does things like calculate the delays, and the lower-level loop 3, which performs the self-calibration imaging. Most of the book-keeping involves handling HDF5 files in some respect. As such, there are a few routines which have been written in Python that perform different operations on HDF5 files. There are four in total and they are described below in turn. These functions are at [https://github.com/mooneyse/lb-loop-2/blob/master/hdf5\\_functions.py](https://github.com/mooneyse/lb-loop-2/blob/master/hdf5_functions.py). They will be implemented into the pipeline shortly.

## 2 Details of the routines

### 2.1 `evaluate_solutions`

#### Description

1. Get the direction from the HDF5 file.
2. Evaluate the phase solutions in the HDF5 file for each station using the XX-YY statistic. This coherence metric is  
`np.nanmean(np.gradient(abs(np.unwrap(phasesol_xx - phasesol_yy))) ** 2)`
3. Determine the validity of each XX-YY statistic that was calculated by checking it is above or below some given threshold.
4. Append the right ascension, declination, and validity to the master text file.

#### Inputs

Usage: `evaluate_solutions(h5parm, mtf, threshold = 0.25)`. The arguments are described below:

- `hdf5` (*string*) – LOFAR HDF5 parameter file.
- `mtf` (*string*) – The path and file name of the master text file. The master text file contains a list of the HDF5 files, the direction of that HDF5 file, and whether its

solutions were deemed good for each station. For example, see <https://github.com/mooneyse/lb-loop-2/blob/master/mtf.txt>.

- **threshold** (*float, optional*) – Threshold to determine the goodness of the XX-YY statistic, which is on a scale of 0 to 1 where 0 is the best situation.

## Outputs

- **None** – This function returns nothing but it will have written results to the master text file.

## 2.2 make\_h5parm\_multiprocessing

### Description

1. This is a wrapper for the `make_h5parm` function, where it runs that function in parallel.
2. This wrapper takes as input a list of arguments to be passed to the `make_h5parm` function, where each list entry is a tuple of parameters for one thread.
3. The `make_h5parm` function gets the direction from the measurement set or takes it from the list provided.
4. It then reads the directions of the h5parms from the master text file.
5. It calculates the separation between the measurement set direction and the HDF5 files directions.
6. For each station, it finds the HDF5 file of smallest separation which has valid phase solutions.
7. It creates a new HDF5 file.
8. It writes these best phase solutions to this new HDF5 file.

### Inputs

Usage: `make_h5parm_multiprocessing(args)`. The arguments are a list of tuples for multiprocessing, and the tuple arguments are described below:

- **mtf** (*string*) – The path and file name of the master text file. The master text file contains a list of the HDF5 files, the direction of that HDF5 file, and whether its solutions were deemed good for each station. For example, see <https://github.com/mooneyse/lb-loop-2/blob/master/mtf.txt>.
- **ms** (*string*) – This is the measurement set which is going to be put through the self-calibration imaging loop (loop 3).

- `clobber` (*bool*, *default = false*) – If true, this will overwrite the new HDF5 file that is being created, should it exist already.
- `directions` (*list*, *default = []*) – Right ascension and declination of the source, in radians.

## Outputs

- The `make_h5parm` function returns the name newly created HDF5 file (i.e. a string).

## 2.3 applyh5parm

### Description

1. It creates an NDPPP parset which will apply the HDF5 file that was outputted by `make_h5parm` to the measurement set.
2. It executes the NDPPP process.

### Inputs

Usage: . The arguments are described below:

- `new_h5parm` (*str*) – This is the output of the `make_h5parm` function.
- `ms` (*string*) – This is the measurement set which is going to be put through the self-calibration imaging loop (loop 3). The HDF5 solutions will be applied to it.
- `clobber` (*bool*, *default = False*) – Whether or not to overwrite the parset, if it already exists.
- `column_out` (*str*, *default = 'DATA'*) – Which column in the measurement set will NDPPP write to.

### Outputs

- This function returns the name of the measurement set as a string which has had the solutions applied to it.

## 2.4 updatelist

### Description

1. Combine the phase solutions from the initial HDF5 file and the final HDF5 file.
2. It calls `evaluate_solutions` to update the master text file with a new line appended.

## Inputs

Usage: `updatelist(new_h5parm, loop3_h5parm, mtf, clobber = False, threshold = 0.25)`. The arguments are described below:

- `new_h5parm` (*str*) – The initial HDF5 file (i.e. the output from `make_h5parm`).
- `loop3_h5parm` (*str*) – This is the final HDF5 file from loop 3.
- `mtf` (*string*) – The path and file name of the master text file. The master text file contains a list of the HDF5 files, the direction of that HDF5 file, and whether its solutions were deemed good for each station. For example, see <https://github.com/mooneyse/lb-loop-2/blob/master/mtf.txt>.

## Outputs

- `combined_h5parm` – A new HDF5 file that is a combination of `new_h5parm` and `loop3_h5parm`.