1 Factor: Facet Calibration for LOFAR[1]

Factor is a Python package that allows for the production of wide-field HBA LOFAR images using the facet calibration scheme described in van Weeren et al. (2015). Note that Factor is still beta software. Please report bugs to `drafferty@hs.uni-hamburg.de`. To initialize your environment for Factor, users on CEP2 and CEP3 should run the following commands:

```
use Lofar
source ~rafferty/init_factor
use Casa
use Pythonlibs
```

1.1 Usage

Factor is run from the command line with the `runfactor` executable as follows:

```
Usage: runfactor [-v|-q|-d] <parset>

Options:
  --version   show program's version number and exit
  -h, --help  show this help message and exit
  -q          enable quiet mode
  -v          enable verbose mode
  -d          enable dry-run mode
```

The parset specifies the parameters of the run. These are described in the next sections.

1.1.1 Quick-start

Below is a quick-start guide to a typical run.

1. Collect the input bands (described in Section 1.3) in a single directory. The MS names can be anything but must end in ".MS" or ".ms":

   ```
   [/data/factor_input]$ ls
   band1.MS        band2.ms        A.MS        b.ms
   ```

2. Edit the Factor parset (described in Section 1.2) to fit your reduction and computing resources. For example, the parset for a single machine could be:

   ```
   [/data]$ more factor.parset
   [global]
   dir_working = /data/factor_output
   dir_ms = /data/factor_input
   parmdb_name = instrument_ap_smoothed
   iteractive = True

   [directions]
   flux_min_jy = 0.1
   size_max_arcmin = 3.0
   separation_max_arcmin = 7.5
   max_num = 30
   groupings = 1:0
   check_edges = True
   ```

---

[1]This section is maintained by David Rafferty (`drafferty@hs.uni-hamburg.de`).

```
    [cluster]
    ncpu = 5
```

3. Run the reduction:

```
[/data]$ runfactor factor.parset
```

1.2 Parset

Below is an template parset that shows the available parameters. Note that the parameters are grouped by sections (denoted by headings such as [global], [directions], etc.). Required parameters are noted.

```
[global]
# Full path to working dir where Factor will run (required). All output will
# be placed in this directory
dir_working = /data/wdir

# Full path to directory containing selfcal-ed bands, will be scanned for all .MS
# and .ms files (required). Note that these files will be modified by Factor, so
# please keep a copy of these data elsewhere
dir_ms = /data/bands

# Parmdb name for dir-indep. selfcal solutions (stored inside the input
# band measurement sets, so path should be relative to those; default =
# instrument)
parmdb_name = instrument_ap_smoothed

# Make final mosaic (default = True)
make_mosaic = True

# Use interactive mode (default = False). Factor will ask for confirmation
# of internally derived DDE calibrators and facets
interactive = False

# Max number of bands per WSClean image when wide-band clean is used (default = 5).
# Smaller values produce better results but require longer run times. Wide-band
# clean is activated when there are more than 5 bands
wsclean_nbands = 2

# Exit if selfcal fails for any direction (default = True). If False, processing
# will continue and the failed direction will receive the selfcal solutions of
# the nearest successful direction
exit_on_selfcal_failure = True

# Use WSClean or CASA for imaging of entire facet (default = wsclean). For large
# bandwidths, the CASA imager is typically faster
facet_imager = casa

# Use chgcenter to minimize number of w planes (default = False). Note that a
# copy of the full dataset will be made (as chgcentre modifies the data)
# NOTE: this option is currently disabled pending testing
use_chgcentre = False
```

```
[directions]
# Full path to file containing calibrator directions. If not given, directions
# are selected internally using the flux and size cuts below
directions_file = /data/directions.txt

# Flux and size cuts for selecting directions internally (min flux, max size
# of a source, and max separation between sources below which they are grouped
# into one direction; required if no directions_file is given). The number of
# internally derived directions can be limited to a maximum number
# of directions if desired with max_num (default = all). These parameters will
# determine the faceting of the field
flux_min_Jy = 0.1
size_max_arcmin = 3.0
separation_max_arcmin = 6.0
max_num = 50

# Check whether any sources from the initial subtract sky model fall on facet
# edges. If any are found, the facet regions are adjusted to avoid them (default
# is False)
check_edges = False

# Total number of directions to process (default = all). If this number is
# greater than ndir_selfcal, then the remaining directions will not be selfcal-
# ed but will instead be imaged with the selfcal solutions from the nearest
# direction for which selfcal was done
ndir_total = 10

# Total number of directions to selfcal (default = all)
ndir_selfcal = 5

# Grouping of directions into groups that are selfcal-ed in parallel, defined as
# grouping:n_total_per_grouping. For example, groupings = 1:5, 4:0 means two
# groupings are used, with the first 5 directions put into groups of one (i.e.,
# each direction processed in series) and the rest of the directions divided
# into groups of 4 (i.e., 4 directions processed in parallel). Shuffling is done
# between neighboring groups to achieve the largest min separation. Default is
# one at a time (i.e., groupings = 1:0)
groupings = 1:5, 4:0

# A target can be specified to ensure that it falls entirely within a single
# facet. The values should be those of a circular region that encloses the
# source and not those of the target itself. Lastly, the target can be placed
# in a facet of its own. In this case, it will not go through selfcal but will
# instead use the selfcal solutions of the nearest facet for which selfcal was
# done
target_ra = 14h41m01.884
target_dec = +35d30m31.52
target_radius_arcmin = 10.2
target_has_own_facet = False


[cluster]
# Full path to cluster description file. Use clusterdesc_file = PBS to use the
# PBS / torque reserved nodes. If not given, the clusterdesc file for a single
```

```
# (i.e., local) node is used
clusterdesc_file = PBS

# Maximum number of CPUs per node to use per node (default = all)
ncpu = 6

# Maximum fraction of the total memory per node that WSClean may use (default =
# 0.9)
fmem = 0.5

# Number of directions to process in parallel on each node (default = 1). If
# directions are split into groups to be processed in parallel (with the
# groupings parameter), this parameter controls how many directions are run
# simultaneously on a single node. Note that the number of CPUs (set with the
# ncpu parameter) will be divided among the directions on each node
ndir_per_node = 1


# MS-specific parameters (optional)
[ms1.ms]
init_skymodel = /data/ms1.sky

[ms2.ms]
init_skymodel = /data/ms2.sky
```

1.3 Data Preparation

The input data must have the average amplitude scale set and average clock offsets removed. Furthermore, the LOFAR beam towards the phase center should be removed. The data should then undergo direction-independent, phase-only self calibration, and the resulting solutions must be provided to Factor. If desired, all sources can be subtracted before Factor is run, thereby allowing one to skip the initial subtraction operation. These steps can be automated using the facet calibration preparation pipeline.

1.4 Directions

The directions for which DDE calibration is performed can be selected automatically in Factor or specified in a file. The format of the file is as follows:

```
# This is an example directions file for Factor.
#
# Directions should be sorted in order of reduction (e.g., bright to faint).
#
# name position atrous_do mscale_field_do cal_imsize solint_ph solint_amp
# field_imsize dynamic_range region_selfcal region_field peel_skymodel
# outlier_source

s1   14h41m01.884,+35d30m31.52 False False 512  1 30  5600 LD empty empty    s1.skymodel True
s2   14h38m29.584,+33d57m37.82 False False 1024 1 30  5600 LD empty empty    empty       False
s25  14h21m07.482,+35d35m22.87 False False 1280 2 240 5600 LD empty s25.rgn  empty       False
```

This file should be specified in the parset file as the `directions_file` parameter.

If no `directions_file` is given, the directions and their order of processing are determined internally using the flux, size, and separation limits specified in the parset as follows. First, the initial apparent-flux clean-component (CC) skymodels (either given as input in the parset or generated by the initial subtraction operation) are grouped by LSMTool using the thresholding algorithm. These grouped patches of CCs are then filtered to meet the specified

flux and size limits (`flux_min_Jy` and `size_max_arcmin`, respectively) and nearby patches are merged (specified by `separation_max_arcmin`). The resulting patches are then sorted by apparent flux (from bright to faint) at the lowest available frequency, and the final `directions_file` is written to the file `factor_directions.txt` in the working directory.

Lastly, one can specify that more than one direction be processed in parallel with the groupings parameter. The groupings parameter specifies the grouping level (the number of directions to process simultaneously) and the total number of directions at each grouping level. For example, `groupings = 1:5, 4:0` means two groupings are used, with the first 5 directions put into groups of one (i.e., the directions are processed in series) and the rest of the directions divided into groups of 4 (i.e., 4 directions processed in parallel). Shuffling is done between neighboring groups to achieve the largest minimum separation between directions in the groups. This sorting attempts to minimize the effects that any artifacts from one direction might have on the other simultaneously processed directions.

1.5 Parallelization

Factor has been designed to run as much as possible in a parallel manner. To this end, support is present for distributing the reduction over the processors and nodes of a compute cluster. To use Factor on a single computer, no setup is necessary beyond specifying the maximum number of CPUs to use. To use Factor on multiple nodes of a compute cluster, the required setup depends on the cluster layout (distributed vs. shared file system) and scheduling system (if any), as described in the following sections.

1.5.1 Cluster without a scheduler

For use on a cluster that does not use a scheduler to assign nodes to the user, such as CEP2 and CEP3, you must provide a LOFAR pipeline cluster description file that tells Factor which nodes are available. An example of such a file is shown below.

```
ClusterName = MyCEP3
Compute.Nodes = [lof021, lof022]
```

1.5.2 Cluster with Torque / PBS

For use on a cluster that uses torque and PBS, you can set `clusterdesc = PBS`. Factor will then automatically determine the nodes for which you have a PBS reservation and use them. Note that you must ask for all the nodes you need in a single PBS script (e.g., to use 6 nodes, with `#PBS -l nodes=6:ppn=6`), so that all nodes are available for the full Factor run. An example PBS script is shown below.

```
#!/bin/bash
#PBS -N Factor
#PBS -l walltime=100:00:00
#PBS -l nodes=6:ppn=6
#PBS -j oe
#PBS -o output-$PBS_JOBNAME-$PBS_JOBID
cd $PBS_O_WORKDIR
source /home/sttf201/init-lofar.sh
runfactor factor.parset
```

1.5.3 Cluster with a shared file system

Factor works by default on a compute cluster with a shared file system.

1.5.4 Cluster with a distributed file system

Factor does not yet work on multiple nodes with a distributed file system (such as CEP2 and CEP3). It does work on a single node of such a system.

1.6   Resuming

Due to the potentially long run times and the consequent non-negligible chance of some unforeseen failure occurring, Factor has been designed to allow easy resumption of a reduction from a saved state and will skip over any steps that were successfully completed previously. In this way, one can quickly resume a reduction that was halted (either by the user or due to some problem) by simply re-running Factor with the same parset.

For example, one can specify that only the first 5 directions be processed. Once these directions are done, Factor will exit. One can then alter the parset and specify that 10 directions should be done. Upon restarting, Factor will skip over the first 5 directions and start with the 6th one (and ending with the 10th one).

1.7   Structure

Factor is structured based on operations and actions. An operation is composed of a sequence of actions. For example, the initial subtraction operation (denoted `initsubtract`) performs the following actions: image at high resolution, make a sky model, subtract the sky model, average, image at low resolution, make another sky model, and finally subtract the second sky model. Operations divide the reduction into logical groupings of actions as follows.

`initsubtract` Performs the initial imaging and subtraction of sources to create "source-free" datasets. This operation may be skipped if sky models are specified for each band in the parset and the `SUBTRACTED_DATA_ALL` column is present in all the input bands.

`facetadd` For each direction, the facet sources are added to the "source-free" datasets. The data are then phase shifted to the appropriate coordinates.

`facetselfcal` The direction-independent calibration is applied, and the bands are concatenated and averaged. Self-calibration is then performed on one or more directions, using two rounds of phase-only calibration followed by two rounds of fast-phase and slow-amplitude calibration. For each direction, all sources in the facet are added to the "source-free" datasets. The data are then phase shifted to the appropriate coordinates. The direction-dependent calibration is applied, and the bands are concatenated and averaged. The entire facet is then imaged, producing a final model of the facet sources.

`facetsub` The final model is subtracted, producing improved "source-free" datasets which are used by subsequent directions.

The facet operations (`facetadd`, `facetselfcal`, and `facetsub`) are repeated for each direction. The `facetselfcal` operation can run on multiple directions simultaneously.

The results of the processing are stored in the `results` directory in the working directory. Inside these subdirectories, results are organized by operation and direction. For example, for a direction named "dir1", the results produced during self calibration are stored in `results/facetselfcal/dir1/`.