Ilifu MeerKAT Open Time Project Guidelines

This document serves to provide guidelines for processing MeerKAT Open Time project data on the Ilifu cloud computing research facility. This document is intended to accompany the *Ilifu Processing and Storage for MeerKAT Projects* and *Ilifu MeerKAT Open Time Project Policies* documents, and outlines guidelines and recommended practices for data management and data processing on the ilifu cluster. This document is written in the context and spirit of supporting research and development. It is understood that there is great variability in project requirements and these guidelines are not intended to hamper research efforts. However, the environment in which we conduct our research is resource-limited, and therefore practices that make efficient use of those resources will only further our ability to successfully support the diverse research teams on Ilifu. To that end, we provide the following guidelines for projects processing and conducting research related to MeerKAT data. It is recommended that MeerKAT Open Time projects develop a data management plan within their project workgroup, informed by these guidelines, to manage data during processing.

Data management

Data processing of MeerKAT data with the IDIA pipeline produces visibility data at several distinct stages. The stages are categorised and described as D1-4 and are referenced in the guidelines below. Generally, these categories are broadly applicable and relevant to any generic workflow.

- **D1**: Raw visibility MeasurementSets transferred from the SARAO archive, without selection and averaging.
- **D2.1**: Pre-processed visibility data that has initial (conservative) flagging for RFI, before averaging. Two versions may be created: a full-spectral resolution version containing only the two parallel hand polarisations for HI science, and a low-spectral resolution version with all four polarisations for continuum imaging and/or polarimetry.
- **D2.2**: Visibility data partitioned into Multi-MeasurementSets (MMSs) in *N* sub-bands for concurrent processing through the cross-calibration stage. After cross-cal, D2.2 includes the corrected and model data columns, and is inflated by ~2.5x compared to D2.1.
- **D3**: Calibrated visibility data with the sub-bands merged into a single MMS. This data set is significantly smaller than D2.2 as it contains only a single "data" column.
- **D4**: Calibration tables and image data products.

Please refer to the *Ilifu Processing and Storage for MeerKAT Projects document*, Section 4.3, for a more detailed description of the data products (D1-4).

Raw Data

- Raw data (D1) should only be transferred if it will be processed within 1 week of completing the transfer; data not being processed should be removed until a later period of processing.
- When transferring raw data to Ilifu, the following selection and averaging parameters should be considered. All of this can be done by configuring data transfers from the SARAO archive to select or average data via the MVFtoMS config, which pre-processes the data such that D1 effectively becomes D2.1.
    - Raw data should only include the data column, not the corrected or model columns, and no autocorrelations;
    - Raw data for extragalactic HI projects should discard data > 1420 MHz and select HH,VV correlations only; HI projects observing local objects should also discard some of the low end of the band if possible;
    - Raw data for continuum and polarisation projects should be averaged to 1k;
    - Raw data for projects with a continuum/polarisation component and HI component should separately transfer 1k data with all four correlations, and 32k data with HH,VV only (product D2.1);
    - Raw data for HI projects that do not need 32k resolution should average between 2-8 channels.
- If the raw data is transferred as D1 (not D2.1), it is recommended to produce D2.1, according to the above selection and averaging parameters, and remove D1 immediately. Raw data is usually set as read-only. Please notify ilifu support at support@ilifu.ac.za when data transferred from SARAO archive can be removed.

Temporary Data

- Data processing should take place on the scratch file-system that is assigned to this project by the Ilifu support team.
- During data processing, unnecessary data and intermediate data products must be deleted.
- D2.2 should be removed immediately, as it is an intermediate state of the cross-cal.
- Projects that use multiple sub-bands / SPWs and an RFI mask should discard frequencies from their RFI mask that align with their SPW boundaries (i.e. not just flag), assuming this doesn't introduce issue for downstream processing with other software (e.g. DDFacet).
- An individual user's scratch directory should not exceed 20 TB.
- Every effort should be made to minimise the project based storage footprint on scratch storage.
- Data should not be duplicated unnecessarily.

Processed Data

- Processed data should be placed within the relevant project directory, including only final or intermediate data products, without duplication, and can be stored for the duration that the project is supported on ilifu.

- Each MeerKAT Open Time project will be granted an allocation of 50 TB for mid/long-term project storage. This excludes the storage available on the scratch folder for short term processing requirements.

Data Access

- If a project's data is proprietary, a user must not enable read permission on their directories containing data from this project to users outside the project group. Please contact support@ilifu.ac.za to inquire about appropriate methods of sharing data.

Compute Resources

- To optimise efficient use of compute resources we recommend that project data be processed on the SLURM cluster by a stable automated pipeline, such as the IDIA pipeline (full Stokes calibration), CARACal and Oxkat, all of which are available on ilifu.
- Pipeline jobs submitted to the SLURM cluster should efficiently use the CPUs allocated to that job, other than short steps/subroutines that comprise <10% of the wall-time of that job; steps that do not efficiently use the job resource allocation should be split into separate jobs[1].
- Job memory allocation should be set as low as possible, according to the best knowledge of the user, allowing for sufficient headroom to ensure the job does not run out of memory.
- Where possible, projects should profile their jobs / pipelines (e.g. through SLURM sacct) and modify the above according to their best knowledge.

[1] We acknowledge the software may not use resources efficiently, which is usually outside the users' control. However, each step should nominally support the use of the resources allocated.