1.2 multibeam - Module

Module for Multibeam receiver data processing

Description

This module pertains to the processing of Multibeam data.
`Livedata`, the Parkes Multibeam data reduction task, directs and regulates the flow of Multibeam data between the following six independent clients:

```
Reader ->
          Bandpass calibrator
                              -> Data monitor
                              -> Writer
                                               -> Statistics
                                               -> Gridder
```

Data flow is from left to right. All clients other than the reader may be disabled or short-circuited, i.e. any line(s) of the above may be deleted other than the first except that the statistics calculator and gridder are dependent on the writer since they process its output file. However, once started, the gridder may remain active even after the writer has been disabled since it is fed via a special purpose queue.
`Livedata` is started by issuing the command

```
% glish -l 'livedata.g'
```

This starts up the **Livedata** GUI from which operations are controlled. The GUI has popuphelp attached. Consult with your local Multibeam expert for details on the reduction process as required.

- Reader: This client reads data from either an MBFITS format file (the single-dish variant of RPFITS), or an AIPS++ Measurement Set. The input data, which may be selected on a beam-by-beam basis, is packaged as a *Glish* record for the next client in the chain.
- Bandpass calibrator: This client does most of the work in calibrating Multibeam data. It operates in several modes depending on the method of observation:
  - Scanning modes as used in HIPASS and ZOA surveys.
  - Frequency switching modes.
  - Position switching modes where each element of the Multibeam system is pointed at the source in turn.

  - Extended source modes such as used for High Velocity Clouds.

    Each of these modes uses a separate bandpass calibration strategy. These are based on robust statistical estimators, particularly median estimators, which allow rejection of RFI without human intervention.

- Data monitor: The data monitor interfaces to **MultibeamView**, a specially modified version of the Karma Karma package **kview** application. In fact, it invokes **MultibeamView** twice, once for each polarization, to provide two panels displaying frequency versus time with various image enhancement options. This provides visual inspection for each pair of polarizations of the 13 beams, one pair at a time.

- Writer: Data is written to an AIPS++ Measurement Set by the writer client. These measurementsets are converted to SDFITS format for data archiving by a separate offline task.

- Statistics: The statistics client reads through an AIPS++ Measurement Set and computes basic statistical measures for each beam and polarization.

- Gridder: Each Multibeam integration consists of a spectrum taken at a particular point on the sky. The gridder takes the spectra from a collection of bandpass-calibrated measurementsets and interpolates them onto a regular map grid, writing the result to a FITS data cube as it goes. Typically each point in the sky is sampled many times in separate observations and the gridder combines these using robust (median) statistics to automatically eliminate Radio Frequency Interference.

  Being the most computationally intensive client, the gridder is usually used separately for offline data reduction. However, it can be linked into the Livedata reduction chain, though, unlike the other clients, it may run remotely on another machine in a quasi-batch mode.

`LiveData` operates in near-realtime at the Parkes telescope to allow immediate assessment of data quality. However, it may also be used for offline data processing.

Each client is implemented as a C++ program with a separate controlling *Glish* script client (subsequence) which provides an optional GUI. The six script clients are coordinated by another script client which directs and regulates the flow of data between them by means of *Glish* events. This "reducer", the `LiveData` control hub, also maintains the input queue for the gridder.

An additional "scheduler" client is added for interactive use. It provides a queueing mechanism which supplies input to the data reduction pipeline. It provides LiveData's near-realtime capability by automatically discovering MBFITS files as they are written by the Multibeam correlator.

LiveData may be run in batch mode by invoking the reducer directly (usually from a *Glish* script) thus bypassing the scheduler (discuss this with your local Multibeam data processing expert).
