**Dynamic, Rule-based Quality Control Framework for Real-time Sensor Data**

Wade Sheldon, Georgia Coastal Ecosystems LTER, University of Georgia (email: sheldon@uga.edu)

### Introduction

Quality control is a critical component of environmental data management, particularly for data collected by autonomous sensors. Performing quality analysis on high volume, real-time data from sensor networks, flux towers and instrumented platforms is a major challenge, though, and can become a limiting factor in managing these data.

Software developed at the Georgia Coastal Ecosystems LTER Site (GCE Data Toolbox for MATLAB) has proven very effective for quality control of both real-time and legacy data, as well as interactive analysis during post processing and effective for quality control of both real-time and legacy data, as well as interactive analysis during post processing and as well as interactive analysis during post processing and as well as interactive analysis during post processing.

This poster describes the dynamic, rule-based quality control framework provided by this software and illustrates how it can be used for both automated and interactive quality management of sensor data.

### Data Model

Underlying the framework is a comprehensive data model that combines data set and attribute metadata, data values, Q/C rules and Q/C qualifier flags into a highly structured data set. An unlimited number of Q/C rules can be defined for each attribute (column), and data values are intrinsically "shadowed" by qualifier flags throughout all operations.

#### Quality Control Rules

Qualification control rules can be pre-defined in metadata templates and applied to raw data automatically upon loading, or can be defined and revised interactively using a GUI editor. Changing rules for a data column automatically triggers re-evaluation, which sets or clears qualifier flags stored for each value in the data column.

#### Manual Q/C Flag Assignment

Qualifier flags can also be assigned or cleared manually to augment or revise rule-based assignments. Manual flagging can be performed on data plots with the mouse, or on a per-column basis using a spreadsheet-like editor dialog. Flags from multiple source columns can also be propagated to one or more dependent columns to augment or overwrite automatically-assigned flags. Once flags are edited manually rules are locked to prevent automatic recalulation, but rules can be unlocked as well.

#### Display of Flagged Values

Flagged values are automatically highlighted when data are viewed in the GUI data editor (below), and flags can be displayed above values on data plots (above). When data are exported in ASCII or MATLAB format qualifier flags can be instantiated as coded data columns in various formats and column arrangements. Records can also be filtered in the data editor to display only those with qualifier flags assigned to simplify data review and revision.

#### Managing Flagged Values

Qualifier flags are an intrinsic part of the GCE Data Toolbox data model, so many options are provided for working with them. Flagged values can be selectively excluded from any analysis (e.g. aggregation, statistical summaries, plots) or exported data set, eliminating the need to delete flagged values from primary data. If flagged values are deleted, all changes are logged to the metadata by column to provide a record for data users.

When data are re-sampled by aggregation, date-time scaling or binning, flagged and missing values are automatically summarized in the derived data set. New Q/C rules can also be generated automatically to flag derived columns based on number or percent missing or flagged values in the original data. Flags can also be locked automatically when multiple data sets are integrated (i.e. unions) to prevent rule conflicts.

### Implementation Scenarios

This quality control framework can be used in a wide variety of scenarios to process sensor data:

- Interactive use:
  - IM or Q/C analyst imports raw data from a data logger or file system using GUI forms
  - Uses metadata templates as starting point, fine-tunes rules, flags
  - Generates finalized data, stats, derived datasets manually

- Batch processing:
  - Metadata templates are pre-defined for each data source
  - Batches of raw data files serially processed using "batch_import"
  - Metadata templates applied, Q/C flags assigned, processed files saved for subsequent operations automatically

- Automated workflow for networked sensors:
  - Metadata templates pre-defined for each data source
  - Workflow scripts defined using high-level API function calls to retrieve data over the network (HTTP, FTP, CIFS, SOAP, SQL)
  - Scripts run periodically by MATLAB (i.e. using native timers) or triggered by other programs (e.g. LoggerNet, CronTab, Kepler)
  - Processed and/or derived data products transmitted to next workflow component on file system or network

The GCE Data Toolbox requires MATLAB 6.5 or higher to run; however, MATLAB is available for all major operating systems and both code and data are highly portable across platforms. Data set size is only limited by computer memory, and million-record data sets can be processed very effectively on commodity desktop PCs or workstations.

### More Information

For more information about the GCE Data Toolbox for MATLAB or to download the software package visit: http://gce-lter.marisc.uga.edu/public/mtools/data_toolbox.htm

Source code is available for review and customization on request, and collaborations are welcome.

---

**Types of Logical Expressions (where x = column data):**

- simple conditionals: $x<0, x>1000$
- compound conditionals: $x>50 \& x<75$
- math or statistical comparisons: $x > \text{mean}(x)+2\times\text{std}(x)$
- cross-references to data in other columns (by name): $\text{col}_1\_\text{Depth}<0.1$ (in column ‘Salinity’)
- $\text{col}_2\_\text{Mass\_Dry} = \frac{\text{col}_3\_\text{Mass\_Total}}{0.8}$
- custom function calls to advanced algorithms, models: $\text{flag}_{2.2}\_\text{saturation}(...)$
- combinations of all above to build complex rules

Flags codes can be any alphanumeric character, and multiple flags can be assigned to an individual data value. Multi-character flags from other data systems can also be mapped to unique flags for analysis. Flag definitions are automatically included in data set metadata.