2007 LTER IM Meeting Demonstration

Title: Dynamic, rule-based QA/QC tools for real-time sensor data

Author: Wade Sheldon, Georgia Coastal Ecosystems LTER

Abstract:

The volume of monitoring data acquired and managed by LTER sites has increased exponentially over time, thanks to advances in sensor technology and computing power combined with steady decreases in data storage costs. New directions in environmental monitoring, such as sensor networks and autonomous roving sensors, promise to raise the bar even higher. The availability of high frequency, real-time data offers exciting new research opportunities, but QA/QC is often a major challenge with real-time data streams due to poor scalability of many traditional approaches and analysis methods. Software developed at GCE LTER (GCE Data Toolbox for MATLAB) has proven very effective for QA/QC of real-time data, though, as well as interactive QA/QC analysis during post processing and synthesis. This demonstration will provide an overview of the QA/QC framework provided by this software, including: 1) creation of QA/QC rule sets based on value/limit checks, mathematical expressions, attribute cross-references and advanced algorithms; 2) creation of metadata templates for automated application of QA/QC rules during data acquisition; 3) graphical tools for editing flag assignments; 4) propagation of assigned flags to dependent attributes; 5) flag encoding and flagged-value removal on export; 6) automatic summarization of flagged/missing values in metadata; 7) automatic flag assignment during re-sampling/synthesis based on flagged and missing values in primary data. Use of this software in automated data harvesting and work-flow scenarios will also be described.

GCE Data Toolbox Quality Assurance/Quality Control Framework

Introduction

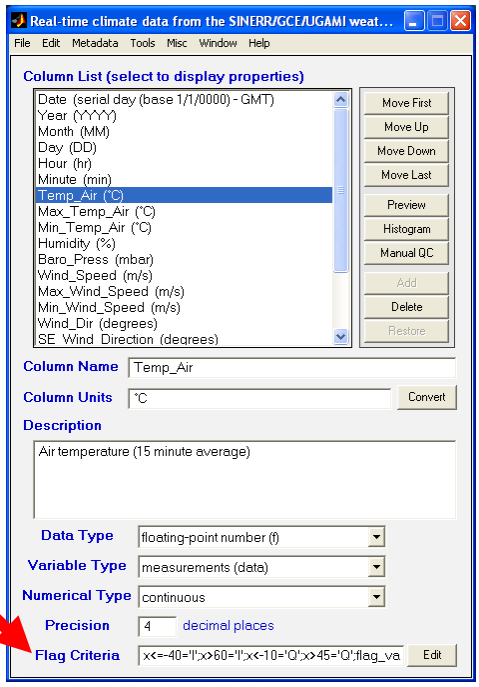
The GCE Data Toolbox for MATLAB® provides a comprehensive framework for Quality Assurance/Quality Control flagging and analysis. In *GCE Data Structures*, the native storage format used by the toolbox (see Appendix fig.9), arrays of data quality "flags" (qualifiers) are created automatically whenever attributes (columns) are added to the structure. These flags are transparently maintained in synchrony with the data they describe throughout all processing steps and analyses. This separation of data values and QA/QC flags obviates the need to delete questionable values from data sets, permitting subsequent re-analysis and flexible handling and display of QA/QC information during analysis and data export.

Flags can be assigned automatically based on QA/QC criteria expressions (i.e. rules) defined for each data column, assigned manually in a spreadsheet-like data editor, or assigned graphically by selecting data points with the mouse. Criteria expressions can include simple conditionals, mathematical formulae and references to built-in or custom MATLAB functions in any combination. Criteria can also include cross-references to other data columns, and flags from multiple columns can be combined and propagated to dependent columns allowing users to perform QA/QC based on complex, multi-column dependency relationships (e.g. flagging of all measured values when a hydrographic instrument is out of the water, based on depth reading).

Flagging of invalid or questionable values in data sets is an important aspect of data processing and management, so QA/QC criteria should be defined whenever practical.

Automatic QA/QC Flagging

Flags can be assigned automatically to values in data columns by defining specific QA/QC criteria (i.e. rules) in the corresponding attribute metadata field (i.e. "Flag Criteria" in the figure to the right). QA/QC criteria are MATLAB® expressions that define alphanumeric flag characters to associate with column values that match the conditions specified. Basic QA/QC criteria (e.g. range or limit checks) can be defined using simple conditional statements, such as "x<0" or "x>=10", where x is a placeholder for the column values. Criteria can also reference any MATLAB® statement or built-in function that returns a logical index of zeros and ones (i.e. zero for no flag, one for flag) or numerical index specifying flags to assign by array position (examples below). Custom QA/QC functions can also be referenced to assign flags based on advanced computations (e.g. statistical analysis, signal processing, time-series analysis), as long as a single logical or numerical index is returned from the function as the first output parameter. A variety of



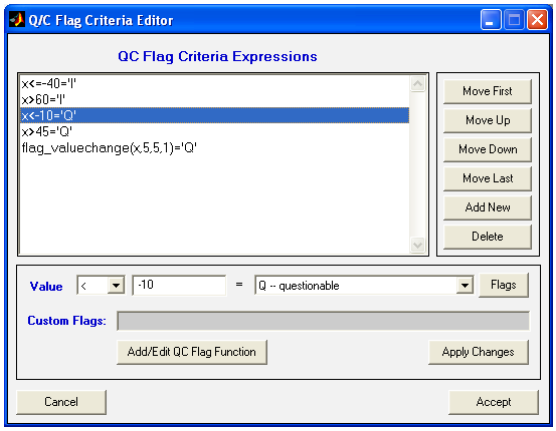
specialized QA/QC functions are provided with the GCE Data Toolbox distribution, and additional functions can be added at any time and referenced in criteria.

Criteria expressions can also include cross references to other data columns, both in conditional statements and function calls, allowing complex dependency-based criteria to be defined. Column references are indicated by prefacing the respective column name with "col_" (e.g. "col_Salinity" to reference "Salinity"). The "col_" prefix can be used in place of "x" for the primary data column reference, if desired, to improve readability of criteria expressions in metadata. Note that missing values in any dependent column will cause the criteria expression to return 0 (no flag) for that value, and incorrect column name spellings or deletion of a referenced column will cause the entire expression to be skipped; however, changes to column names and units performed in the Data Structure Editor and using toolbox functions will automatically be propagated to all flag criteria expressions in the data set to maintain validity of QA/QC criteria and dependencies.

Note that QA/QC criteria defined in metadata templates are evaluated automatically whenever the template is applied to a dataset. Defining criteria in templates is therefore a powerful mechanism for performing automatic QA/QC for newly acquired or harvested raw data (see *Automated QA/QC in Scripted Batch-mode Scenarios*). Criteria are also re-evaluated automatically whenever criteria or data values are updated using GCE Data Toolbox programs, unless flags are locked by insertion of the "manual" token (see *Manual QA/QC Flagging*).

QA/QC Criteria Syntax and Examples

Flag criteria expressions follow the pattern [*condition*]=[*flag code*], where [*condition*] is any MATLAB expression (or function call) that returns a logical or numerical index, and [*flag code*] is a corresponding alphanumeric flag code to assign when the condition is met. A GUI criteria editor is provided in the GCE Data Toolbox to simplify defining, editing and re-ordering Q/C criteria expressions (figure on right). This editor can be invoked by pressing the "Edit" button next to the criteria field on the Data Editor window (above).



Specific syntax and examples are listed below:

1) Numeric conditionals (e.g. limit/range checks):

Syntax: x[operator][value]='[flag]' , where:
x (or col_[column name]) is an alias for values in the current data column
[operator] is ==, <, >, <=, >=, ~= (or <>)
[value] is a numeric value (scalar or array the same size as "x")
[flag] is any one text character, symbol, or digit enclosed in single quotes

Examples:

x<0='I' -- generates 'I' flags for negative values (e.g. for mass or count data)

x>=30='Q' -- generates 'Q' flags for values 30 or higher

x~=1='Q' -- generates 'Q' flags for values other than 1

2) Column cross-references (e.g. dependency checks):

Examples:

col_Depth<0='I' (in column Salinity) -- generates 'I' flags for salinity values when values in Depth are negative (i.e. instrument out of the water)

col_Dry_Weight>col_Wet_Weight='I' (in column Dry_Weight) -- generates 'I' flags for dry weights that exceed the total wet weight for a sample

3) Basic mathematical expressions (e.g. multi-column dependency checks):

Example:

col_Wet_Weight>(col_Dry_Weight+col_Water_Weight)='Q' (in column Wet_Weight) -- generates 'Q' flags for wet weights that exceed dry weight plus water weight (note that parenthesis can be used to control order of operations)

4) Built-in MATLAB numeric functions (e.g. missing value or statistical checks):

Examples:

isnan(x)='M' -- generates 'M' flags for any missing numerical values (NaN)

x<(mean(x)-3.*std(s))='Q' -- generates 'Q' flags for any values < 3 standard deviations below the column mean (assumes no missing values)

x<(mean(x(~isnan(x)))-3.*std(x(~isnan(x))))='Q' -- same as above, allowing for missing values

5) Built-in MATLAB string functions (e.g. code checks):

Examples:

strcmp(x,'none')='M' -- generates 'M' flags for strings matching 'none'

~strcmp(x,'none')='G' -- generates 'G' flags for strings not matching 'none'

strncmpi(x,'Spartina',8)='G' -- generates 'G' flags for strings with the first 8 characters matching 'Spartina', ignoring case

cellfun('isempty',x)='M' – generates 'M' flags for missing values (empty strings)

6) Custom MATLAB functions (single column criteria):

Any MATLAB function that accepts column values as input and returns a logical or numeric index as its first output variable can be used in criteria. Note that a function call editor with syntax help is available from the 'Q/C Flag Criteria Editor' tool.

Examples:

flag_percentchange(x,20,20,3)='Q' -- generates 'Q' flags for any values that vary by more than 20% below or above the mean of the preceding 3 values (note: input parameters are 'value','lowlimit','highlimit' and 'framesize', resp.)

flag_notinlist(col_Plant_Species,{'Spartina','Juncus','Borrichia'})='Q' -- generates 'Q' flags for any values in 'Plant_Species' that are not in the specified list of allowed values (note that external code list files can also be referenced)

7) Custom MATLAB functions (multiple-column criteria):

Same as single-column custom function syntax, except additional column values are entered as function arguments, using the column reference format: col_[column name].

Examples:

flag_o2saturation(col_Oxygen,col_Temperature,col_Salinity,110,50)='Q' -- generates 'Q' flags for any oxygen values that are above 110% saturation or below 50% saturation based on the oxygen saturation calculated as a function of oxygen concentration, temperature and salinity

flag_locationcoords(col_Site,col_Longitude,col_Latitude,0.2, 'gce_locations.mat')='Q' -- generates 'Q' flags for any location names in 'Site' with longitude and latitude values that deviate more than 0.2km from the coordinates registered in 'gce_locations.mat' by dead reckoning (i.e. flags geo-referencing errors in data sets)

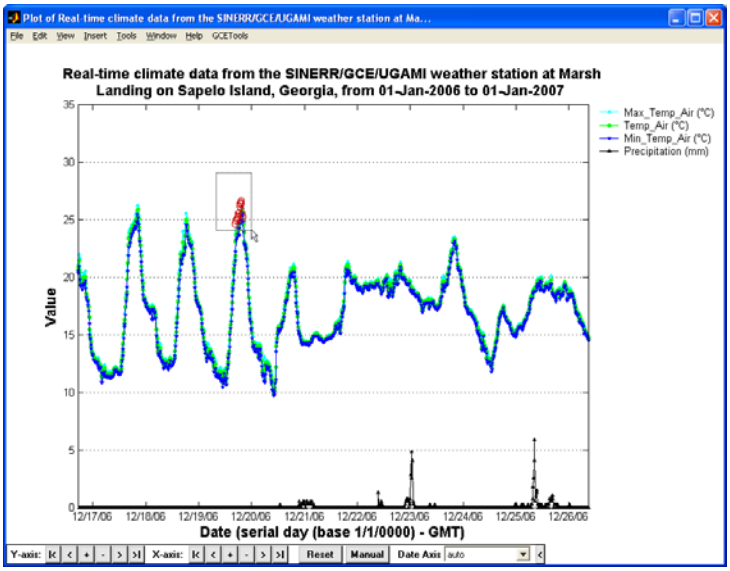
8) Compound criteria:

Multiple criteria can be specified for each column by using a semicolon to separate each one. Overlapping criteria are supported, resulting in multiple flag assignments when more than one criteria is matched. Note that certain operations (e.g. encoding flags as unique integers - automatic for MATLAB file export) will only retain the first-assigned flag, therefore order of precedence should be considered (e.g. list rules that assign 'invalid' flags before rules that assign 'questionable' flags).

Example:

x<0='I';col_Depth<0.1='I';x>36='Q';flag_percentchange(x,20,20,3)='Q' (in "Salinity") -- generates 'I' flags for negative values, 'I' flags for values recorded when Depth was < 0.1, 'Q' flags for values > 36 and 'Q' flags for values that are 20% above or below the mean of the three preceding values.

Manual QA/QC Flagging



Flags can also be assigned manually using various GCE Data Toolbox programs and utilities. For example, data values displayed on line/scatter plots can be flagged (or un-flagged) with the mouse using the "Visual Q/C Tool" available in plot figure menus. The user just selects a data column and flag to assign, then clicks on individual values or drags a rectangle over a range of values with the mouse (figure on right). Whenever flags are manually assigned or cleared, the token "manual" is appended to the criteria field for the respective data column(s) to lock the flags and prevent automatic recalculation. Automatic flagging can be reinstated by removing the "manual" token from the criteria string or by using the "Unlock Q/C Flags" option under the "Edit > Q/C Flag Functions" menu in the Data Editor window.

Similarly, flags assigned prior to importing data into the GCE Data Toolbox (e.g. flags assigned by a data provider, such as USGS, NOAA, or LTER ClimDB/HydroDB) can also be converted to flag arrays and meshed with (or replace) existing flags assigned by QA/QC criteria or manual editing. Predefined flag fields should be text columns that are named according to the convention "Flag_[column name]", e.g. "Flag_Salinity" for column "Salinity".

Flag Codes and Metadata

QA/QC flag codes should be documented in the metadata (i.e. 'Data' category, 'Codes' field) using the following format: "Q = questionable value, I = invalid value, M = missing", etc. This ensures that the flag codes are properly displayed in standard and XML metadata, and also allows flag code definitions to be automatically generated when flags are converted to encoded integer columns during ASCII or MATLAB export operations (or manually in the Data Editor). A GUI flag definition editor is provided with the GCE Data Toolbox, which can be opened using the 'View/Edit Q/C Flag Definitions' option on the 'Edit > Q/C Flag Functions' menu.

Suggested flag codes are listed below:

I = invalid value (out of range) -- use for out-of-range/impossible values (e.g. mass < 0)
Q = questionable value -- use for values outside of expected range (e.g. below detection limit, well outside of historical value range, pattern indicating data contamination)
E = estimated value -- use for values that were estimated by interpolation or other means
S = spike/noise -- use for sharp discontinuities/spikes indicating data contamination

Automated QA/QC in Scripted Batch-mode Scenarios

The GCE Data Toolbox is well suited to use in scripted batch-mode data processing scenarios. Dataset metadata are used to automatically parameterize toolbox functions, so simple high-level commands can be used to carry out complex multi-step processing and analysis. All operations performed using GUI forms can be accomplished using a corresponding command line statement in a script, including propagation of flags to dependent columns, selective removal of flagged values, and automatic flagging of derived data sets (e.g. aggregated, temporally-resampled and binned data) based on number or percentage of flagged and/or missing values in primary data.

The key to performing automated QA/QC in unattended batch mode is to create a metadata template for the data source, containing appropriate QA/QC criteria (rules) for each attribute. When the template is applied to the raw data after loading or importing, QA/QC flags are automatically assigned to each attribute based on these criteria. The full suite of QA/QC-related functions can then be used to manage the display of flags in exported data products and plots, or to remove values assigned particular flags or perform other operations. Note that a GUI editor is provided with the GCE Data Toolbox for defining, managing and editing metadata templates.

Once a suitable metadata template is defined, simple functions or scripts can be used to fully process raw data files, for example:

```
[s,msg] = imp_ascii('weather.txt','d:\data\met','Weather Data','weather_template');
[s,msg] = clearflags(s,'I');
msg = exp_ascii(s,'tab','weather_qc.txt','d:\data\met','Weather Data','ST','M','FLED');
```

This script would perform the following operations:

1. import and parse a raw ASCII data file (d:\data\met\weather.txt), automatically applying the 'weather_template' metadata template and assigning QA/QC flags after import
2. remove values assigned 'I' flags, converting to NaN (null), retaining other flagged values
3. export the processed data in tab-delimited ASCII format, with column titles, separate metadata file (in ESA FLED style), and text flag columns following the corresponding data columns

Additional commands could also be included to fill in missing records to create monotonic time series, add derived parameters based on equations referencing data columns (each with their own QA/QC criteria), and resample or filter the data to produce derived data products that can be

further manipulated and exported along with the primary data. Specialized import filters can also be defined to perform an entire prescribed workflow using a single command. Such filters are included with the GCE Data Toolbox distribution for USGS NWIS data, NOAA NCDC climate data, LTER ClimDB/HydroDB data, NOAA HADS data and other sources.

Recent versions of MATLAB also include support for timed program execution, network data access (via HTTP, FTP and UNC paths), and a SOAP web services client, allowing the GCE Data Toolbox to be used for automated remote data acquisition and QA/QC processing. At GCE, fully automated data harvesters have been developed for NOAA HADS data, USGS NWIS data, and LTER ClimDB/HydroDB data (i.e. the USGS data harvesting service for HydroDB).

QA/QC Flag Handing in Post Processing

QA/QC flags are a constitutive component of GCE Data Structures (Appendix fig.9), so most GCE Data Toolbox GUI dialogs and functions provide explicit options for handling flagged values in data sets during post processing and analysis. For example, flags can be displayed, ignored or removed when data are plotted, and summary statistics displays and reports can be generated with and without flagged values (or both), and numbers of flagged values are summarized for each attribute. Data export functions also provide various options for formatting flags in delimited ASCII and MATLAB files to support other programs and standards. Data integration tools (e.g. merge/union and join) also provide options for "locking" QA/QC flags to prevent inappropriate application of criteria after multiple data sets are combined.

Data aggregation, date/time re-sampling, and binning tools offer particularly fine-grained control over QA/QC flags. Values assigned specific flags can be removed prior to analysis, and QA/QC criteria can be defined automatically for derived data columns based on the number or percentage of flagged and/or missing values in each respective group, date/time interval or bin. Attributes listing the number (and percentage) of flagged and missing values are also included in derived data sets. Information on the quality and completeness of primary data can therefore be documented and preserved in derived data to guide usage and interpretation.

Additional Information

Information about the GCE Data Toolbox and links to downloadable files are available on the web at: http://gce-lter.marsci.uga.edu/public/im/tools/data_toolbox.htm. Information about MATLAB®, which is required to run the software, is available on the MathWorks web site at: http://www.mathworks.com/.

Note that this software is a core component of the information system at the Georgia Coastal Ecosystems LTER site, and is not currently available under an open source license; consequently, the public distribution of the toolbox contains pre-compiled MATLAB programs (p-files) and documentation-only source files (m-files). Source code will be provided to other LTER sites and LNO personnel on request, provided recipients agree to collaborate on future development and proposals they initiate, and do not distribute the code beyond their collaborative environment without permission. Other requests for source code access will be considered on a case-by-case basis.

Please contact Wade Sheldon (sheldon@uga.edu) for additional information about this software and to discuss potential collaboration opportunities.

Appendix: GUI Application Screenshots and Other Figures

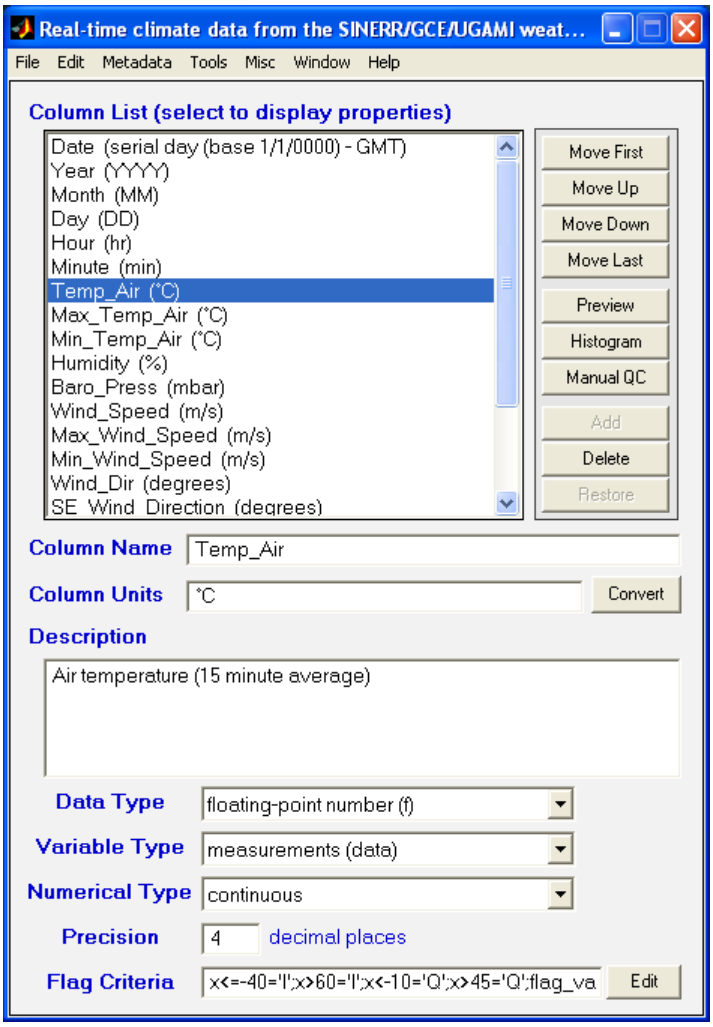


Figure 1. Data Editor dialog, depicting the QA/QC flag criteria and other attributes of the selected data column. Note the 'Manual QC' button in the right panel, which displays the current column and flag array in a spreadsheet-like application for manual editing, and the 'Edit' button to the right of the 'Flag Criteria' field, which opens the Q/C Flag Criteria Editor dialog (fig. 2). Additional QA/QC-related dialogs and options are available under the 'Edit' menu.

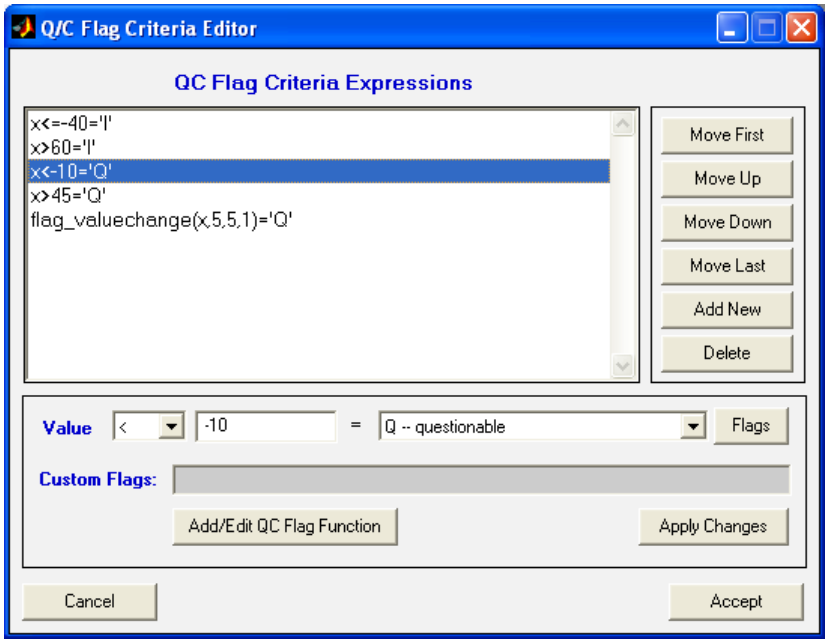


Figure 2. Q/C Flag Criteria Editor window invoked by pressing the 'Edit' button next to the criteria field on the Data Editor window (fig. 1). This dialog is used to create, edit and re-order QA/QC criteria statements stored in the criteria field of each data attribute.

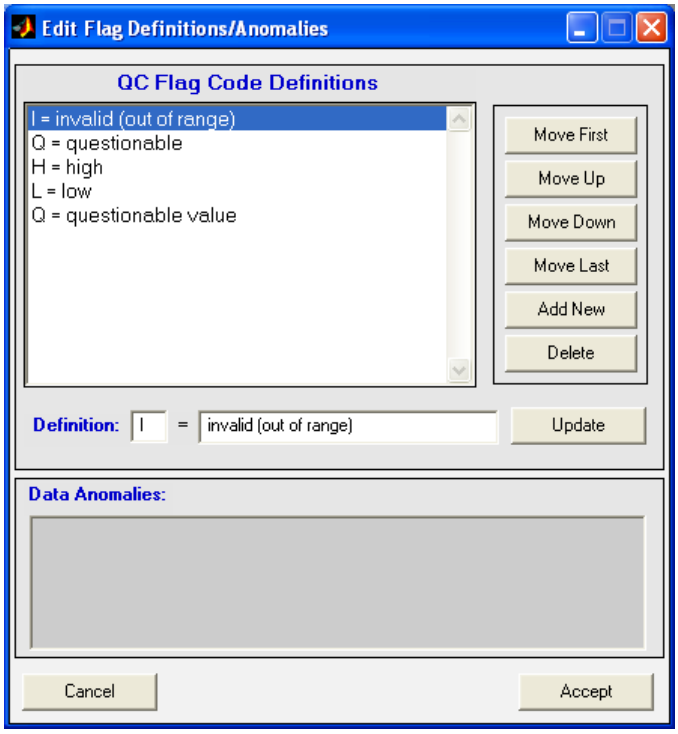


Figure 3. Flag Definition editor window invoked by pressing the 'Flags' button on the Q/C Flag Criteria Editor dialog (fig. 2). This form is used to define flag codes in the data set metadata, which are referenced by all QA/QC functions and dialogs.

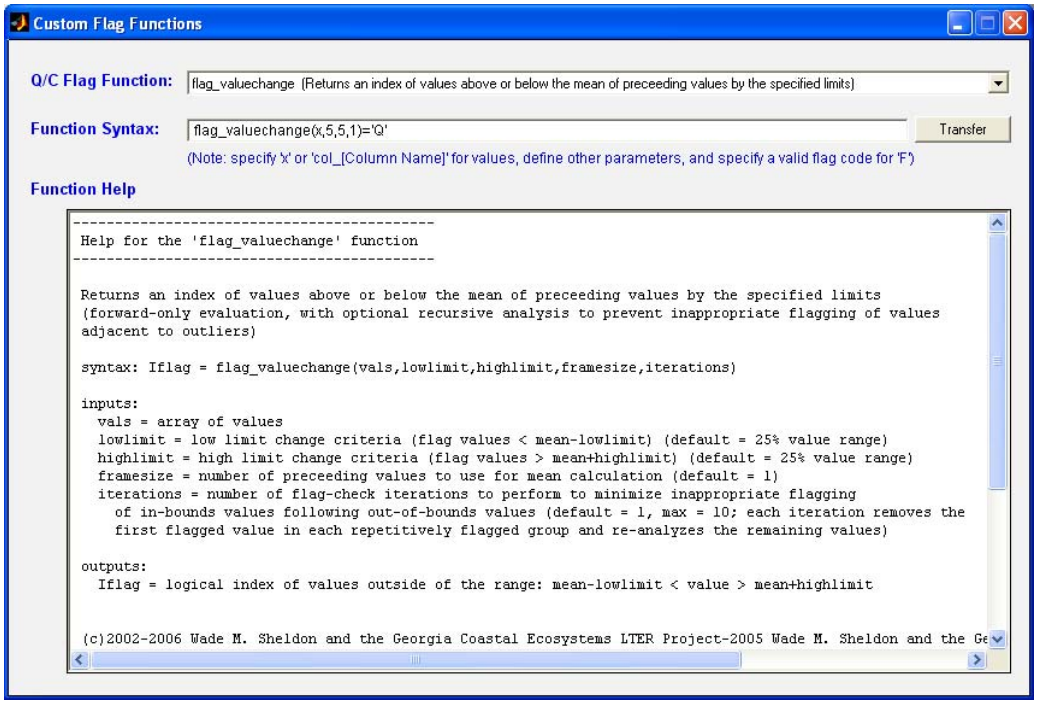


Figure 4. Custom Flag Function editor window invoked by pressing the 'Add/Edit Q/C Flag Function' button on the Q/C Flag Criteria Editor dialog (fig. 2). This form is used to define QA/QC criteria based on custom MATLAB functions. Help text and syntax are provided for all QA/QC functions provided with the GCE Data Toolbox.

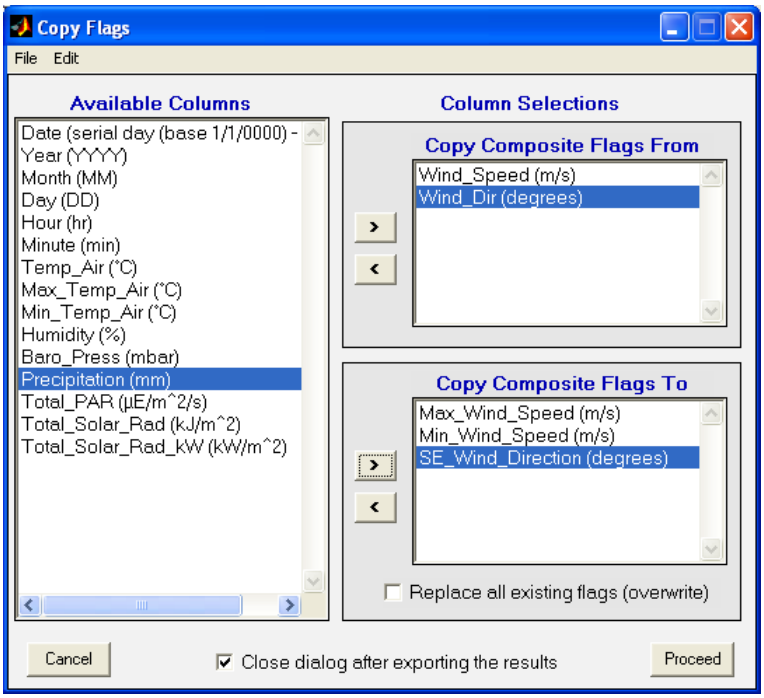


Figure 5. Copy Flags dialog, which supports manually propagating flags assigned to one or more columns to multiple dependent columns (automatically locking QA/QC criteria).

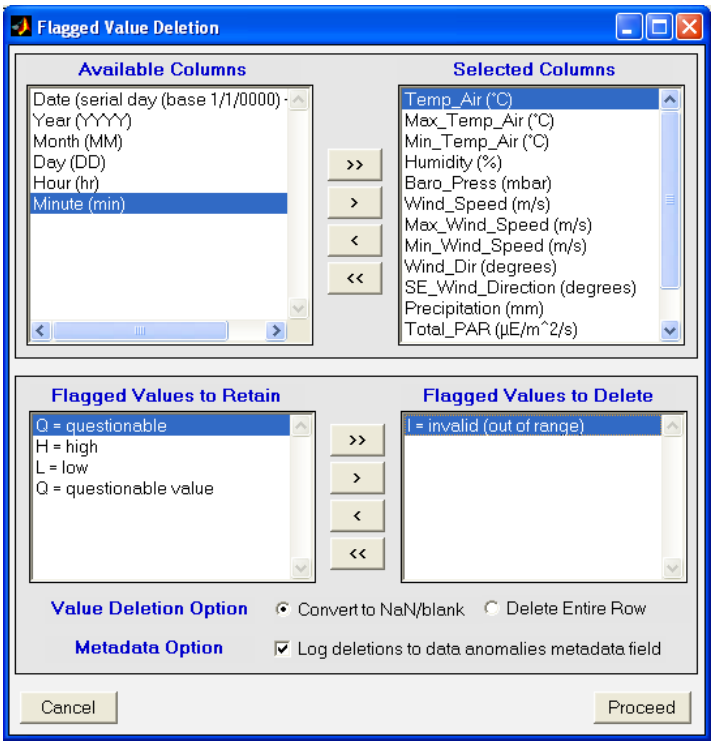


Figure 6. Flagged Value Deletion dialog, which supports selectively removing values (or entire records) from data sets based on QA/QC flag assignments.

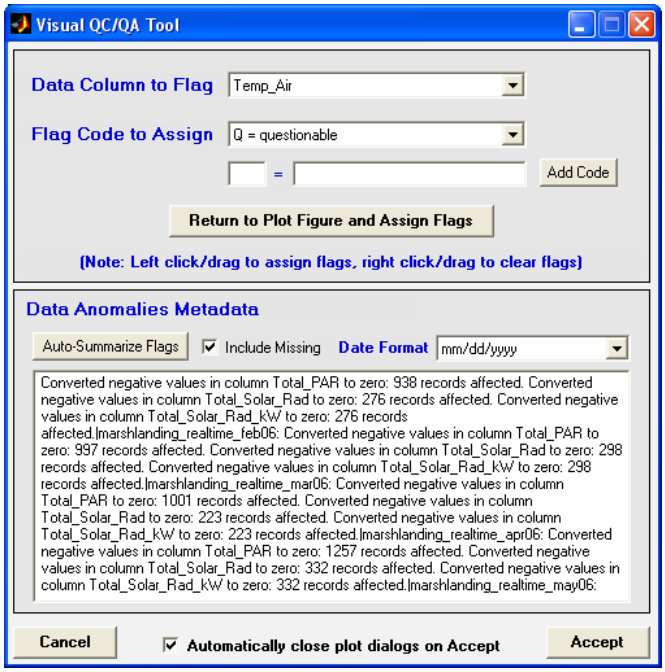


Figure 7. Visual QA/QC flagging dialog for selection of parameters and flags to assign (or clear) via mouse on data plots, plus creation and editing of data anomalies metadata.

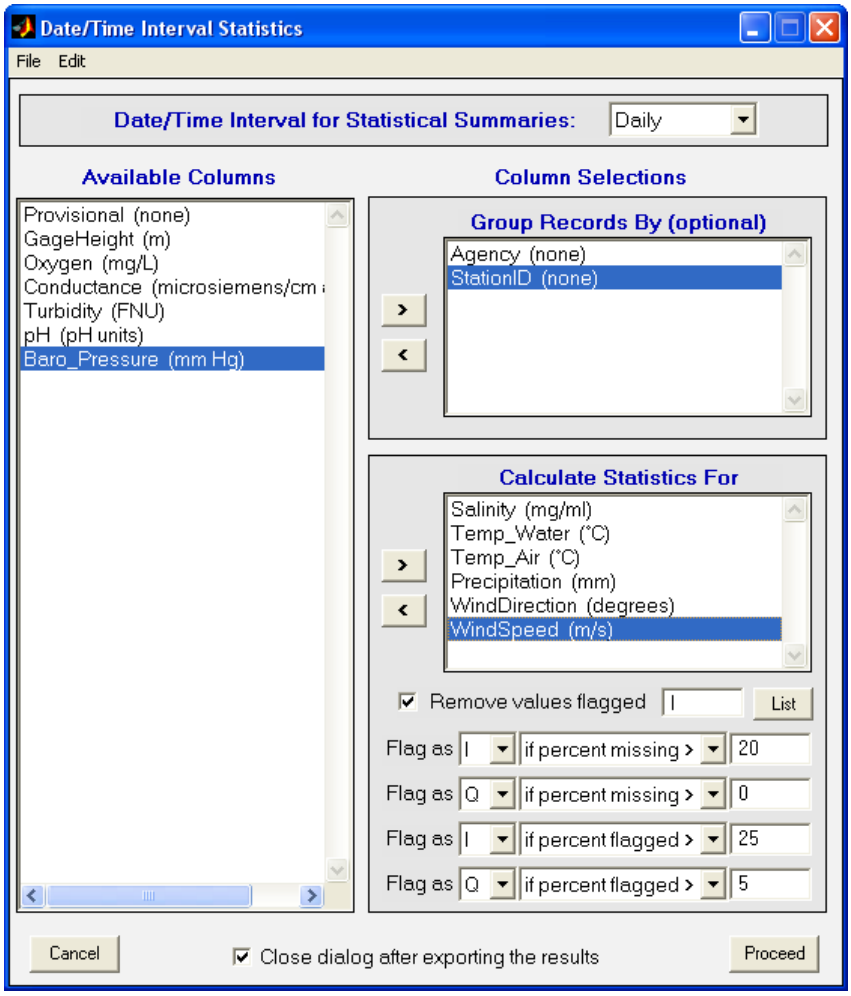


Figure 8. QA/QC flag handing option in the date/time interval statistics dialog. Flagged values can be selectively removed prior to performing aggregation, and QA/QC criteria can automatically be defined for calculated attributes in the derived data set based on the number or percentage of residual flags and/or missing values in the original data set. The "List" button opens a GUI dialog containing a list of all flag codes and definitions in the metadata for selection. The same options are also provided in general aggregation and binned statistics dialogs.

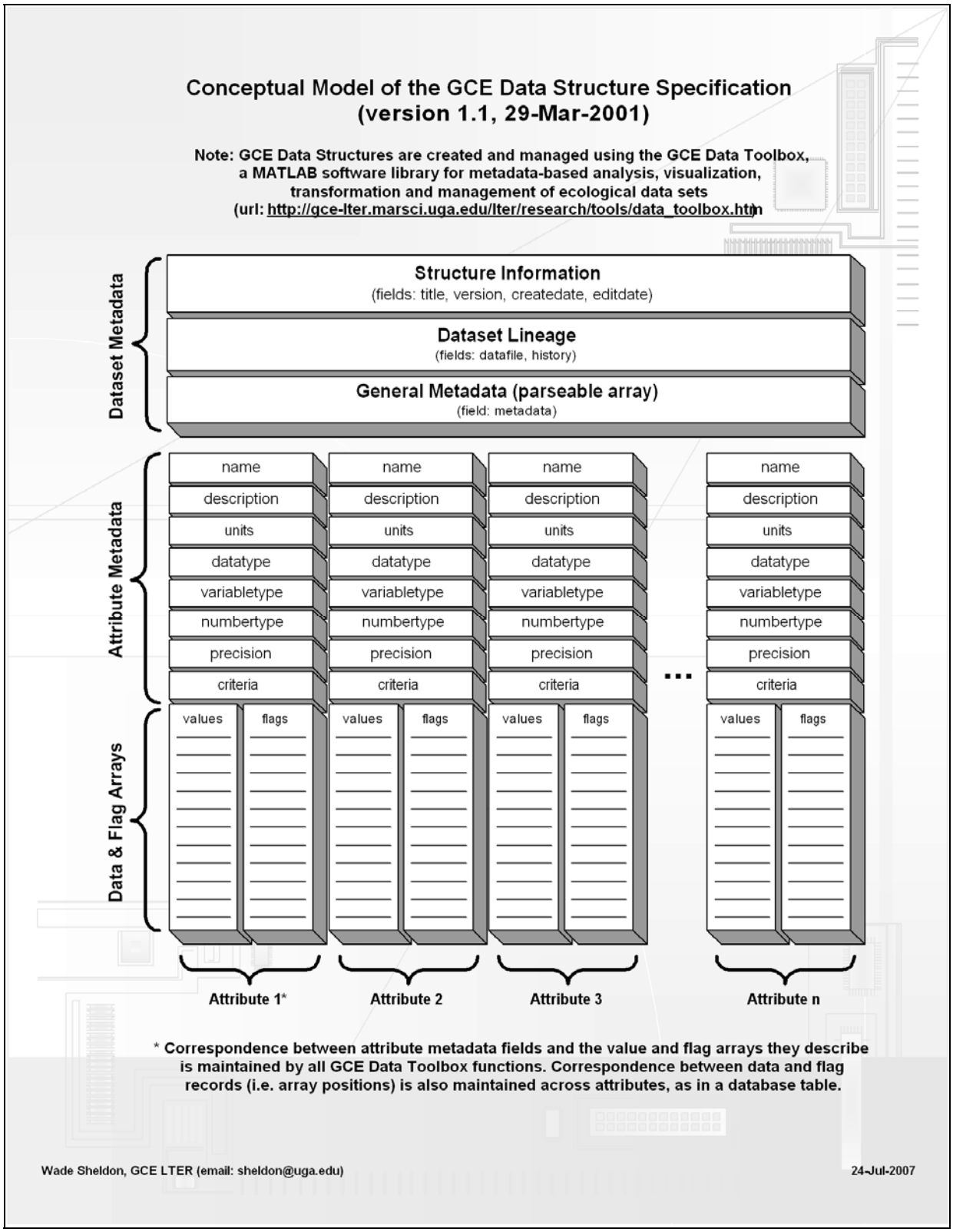


Figure 9. Conceptual model of the GCE Data Structure (implemented as a MATLAB 'struct' array), which is the native storage format used by the GCE Data Toolbox for MATLAB.