

The GCE Data Toolbox Q/C Framework

Wade Sheldon

Georgia Coastal Ecosystems LTER

University of Georgia

email: sheldon@uga.edu

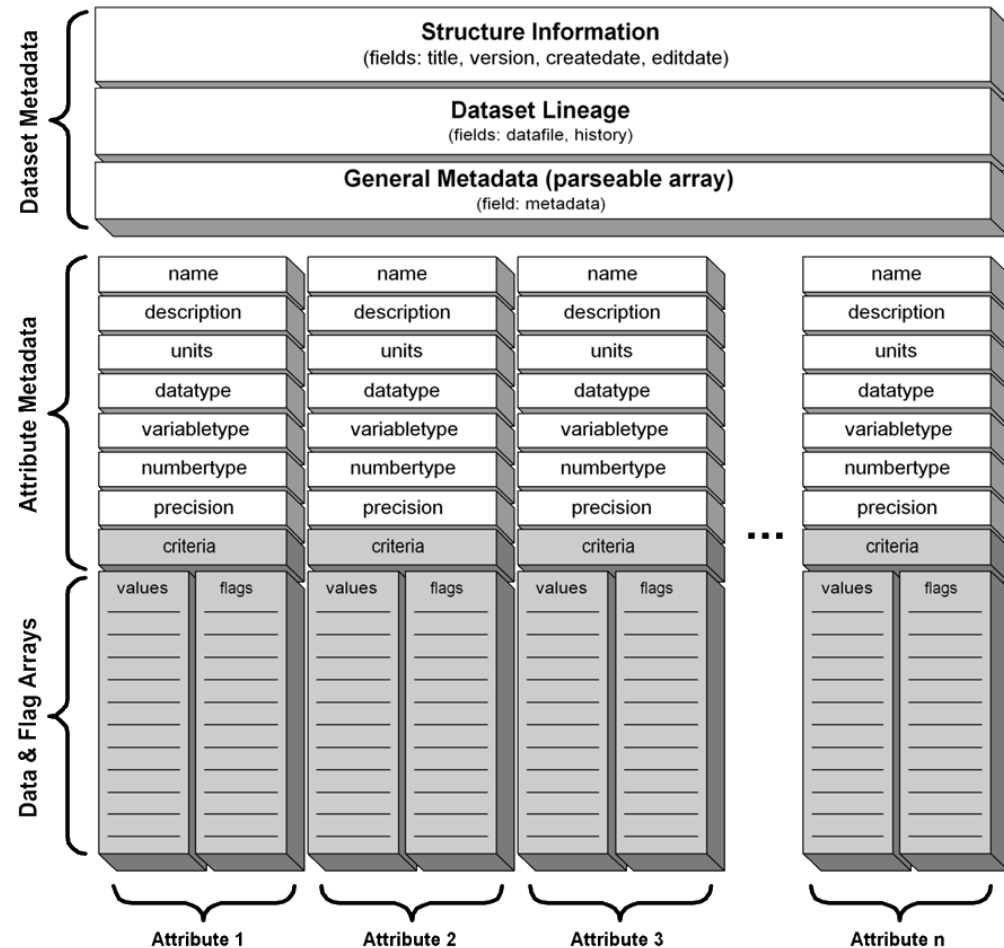


Overview

- ▶ Four major facets of Toolbox Q/C framework
 - Data model for storing Q/C rules, qualifiers, Q/C metadata
 - Tools for performing Q/C (assigning qualifier flags)
 - Tools for managing qualifier flag assignments
 - Tools for visualizing and managing qualified values
- ▶ Components work together seamlessly
 - Q/C framework integrated with data processing, analysis
 - Supports Q/C throughout the entire data lifecycle
- ▶ Q/C strategy important
 - Q/C is an iterative process
 - Operations done in wrong order can nullify flags, waste effort
 - Information can also be lost

Data Model

- ▶ GCE Data Structure designed to store data, Q/C info, metadata together
- ▶ Dedicated Q/C criteria field
- ▶ Paired value/flag arrays
- ▶ Q/C fields in doc metadata
 - Data/Codes (flag definitions)
 - Data/Anomalies (Q/C summary)
 - Data/ValueCodes (definitions for instantiated flag columns)
 - Processing history (all Q/C operations)



Data Model – Q/C Principles

- ▶ Data and structure validation happens before Q/C analysis
 - No need to add Q/C checks for valid data type, ragged tables, text in numeric fields, ...
- ▶ Basic Q/C operations are documented automatically
 - Edits to rules, evaluation of rules, manual/visual Q/C, flagged value removal, ...
- ▶ No qualifier vocabularies or semantics are assumed
 - Qualifiers are defined in the data set metadata (or interactively)
 - Just as valid to assign qualifiers for “good” values as “bad” (e.g. “reviewed”)
 - Subtle bias for “bad” interpretation in some tools, but not hard and fast
- ▶ Multiple qualifiers can be assigned to a data value
 - Multiple Q/C criteria matched
 - Multiple qualifiers assigned/copied manually
 - Major exception is graphical assignment (1 flag set or all flags cleared)

Assigning Qualifiers

- ▶ Qualifiers can be assigned automatically
 - Assigned based on criteria expressions for each column (“rules”)
 - Assigned by specialized flag functions in rules or run in workflows
 - Assigned by tools when values revised, calculated (e.g. gap-filling)
- ▶ Qualifiers can be assigned manually/interactively
 - Typed into a data grid or added using workflow functions
 - Graphically assign with the mouse on data plots
- ▶ Qualifiers can be copied/imported
 - Copy composite flags from one or more columns to other column(s)
 - Convert text columns to flags to import pre-assigned or 3rd party flags

Automatic Assignment (Rules)

- ▶ Basic syntax: [**logical expression**]='**flag code**'
- ▶ Logical Expression =
 - MATLAB expression that evaluates each data value against some criteria and returns true/false (true mean matches criteria)
- ▶ Flag Code =
 - Alphanumeric character to assign when expression true (I, q, 9, *)
 - Flag Codes defined in the dataset metadata (I = invalid value, Q = questionable ...)
- ▶ Examples
 - Conditional statements: $x < 0 = 'Q'$
 - Mathematical/statistical comparisons: $x < \text{mean}(x) - 2 * \text{std}(x) = 'Q'$
 - Functions for set-based comparisons: $\text{flag_notinlist}(x, 'A,B,C') = 'I'$

Automatic Assignment (Rules)

▶ Rule Characteristics

- Column values are referenced in rules using “x” or “col_[name]”
- Rules are evaluated for every value in the column independently
- Rules can reference corresponding values in other columns
- Multiple expressions can be combined using Boolean operators (&/||) and include parentheses to create complex criteria
- Can define any number of rules per column (semi-colon separator)
 - If multiple rules are matched, multiple flags are assigned (e.g. IQQ)
- Can use mathematical statements, MATLAB (or toolbox) functions to create derived arrays for comparison with column values
 - mean(), max(), std(), sum()
- Can call functions to retrieve reference data, run algorithms, etc (open-ended)

Automatic Assignment (Rules)

- ▶ Rules can be created, edited, deleted, re-ordered using GUI forms

The image displays three GUI windows used for creating and managing rules for automatic assignment of data flags.

Real-time climate data from the SINERR/GCE/UGAMI weather station...

Column List (select to display properties)

- NESDIS_ID (none)
- NWSLID (none)
- Date (serial day (base 1/1/0000) - GMT)
- Year (YYYY)
- Month (MM)
- Day (DD)
- Hour (hr)
- Minute (min)
- Temp_Air (°C)
- Max_Temp_Air (°C)
- Time_Max_Temp_Air (hhmm)
- Min_Temp_Air (°C)
- Time_Min_Temp_Air (hhmm)
- Humidity (%)
- Baro_Press (mbar)
- Wind_Speed (m/s)

Column Name: Wind_Speed

Column Units: m/s

Description: Wind speed, 15 minute average

Data Type: floating-point number (f)

Variable Type: measurements (data)

Numerical Type: continuous/ratio (continuous)

Precision: 3 decimal places

Flag Criteria: $x < 0 = 'I'; x > 50 = 'I'; x > 30 = 'Q'; col_Wind_Speed > col_Max_Wind_Speed$

Q/C Flag Criteria Editor

Data Set Column: Wind_Speed (m/s)

Q/C Flag Criteria Expressions

```
x < 0 = 'I'
x > 50 = 'I'
x > 30 = 'Q'
col_Wind_Speed > col_Max_Wind_Speed = 'Q'
```

Data Restriction Query Builder

Data Structure Columns

Column	Operator	Column	Action
NESDIS_ID (none)			Clear
NWSLID (none)			Clear
Date (serial day (base 1/1/0000) - GMT)			Clear
Year (YYYY)			Clear
Month (MM)			Clear
Day (DD)			Clear
Hour (hr)			Clear
Minute (min)			Clear
Temp_Air (°C)			Clear
Max_Temp_Air (°C)			Clear
Time_Max_Temp_Air (hhmm)			Clear
Min_Temp_Air (°C)			Clear
Time_Min_Temp_Air (hhmm)			Clear
Wind_Speed	>	Max_Wind_Speed	Pick

Match Option: ALL **Query:** Wind_Speed > Max_Wind_Speed

Automatic Assignment (Rules)

- Syntax help displayed for custom flag functions to guide use

Real-time climate data from the SINERR/GCE/UGAMI weather station...

File Edit Metadata Tools Misc Window Help

Column List (select to display properties)

- NESDIS_ID (none)
- NWSLID (none)
- Date (serial day (base 1/1/0000) - GMT)
- Year (YYYY)
- Month (MM)
- Day (DD)
- Hour (hr)
- Minute (min)
- Temp_Air (°C)
- Max_Temp_Air (°C)
- Time_Max_Temp_Air (hhmm)
- Min_Temp_Air (°C)
- Time_Min_Temp_Air (hhmm)
- Humidity (%)
- Baro_Press (mbar)
- Wind_Speed (m/s)

Move First Move Up Move Down Move Last Preview Histogram Manual QC Add Delete Restore

Column Name Wind_Speed

Column Units m/s Convert

Description

Wind speed, 15 minute average

Data Type floating-point number (f)

Variable Type measurements (data)

Numerical Type continuous/ratio (continuous)

Precision 3 decimal places

Flag Criteria x<0='I';x>50='I';x>30='Q';col_Wind_Speed>col_Max_Wind_Speed='Q' Edit

Q/C Flag Criteria Editor

Data Set Column: Wind_Speed (m/s)

Q/C Flag Criteria Expressions

```
x<0='I'
x>50='I'
x>30='Q'
col_Wind_Speed>col_Max_Wind_Speed='Q'
```

Move First Move Up Move Down Move Last Add New Delete

Value: [] = [] -- invalid (out of range) Flags

Custom flags: col_Wind_Speed>col_Max_Wind_Speed='Q' Help

Add/Edit QC Flag Function Add/Edit Custom Criteria Cancel Accept

Customizing Functions

Q/C Flag Function: flag_valuechange (Returns an index of values above or below the mean of preceding values by the specified limit)

Function Syntax: flag_valuechange(vals,lowlimit,highlimit,framesize,iterations)='F' Transfer

(Note: use 'X' or 'col_[Column Name]' to pass data values, add additional arguments, and specify a flag code to assign)

Function Help

Help for the 'flag_valuechange' function

Returns an index of values above or below the mean of preceding values by the specified limits (forward-only evaluation, with optional recursive analysis to prevent inappropriate flagging of values adjacent to outliers)

syntax: Iflag = flag_valuechange(vals,lowlimit,highlimit,framesize,iterations)

inputs:

- vals = array of values
- lowlimit = low limit change criteria (flag values < mean-lowlimit) (default = 25% value range)
- highlimit = high limit change criteria (flag values > mean+highlimit) (default = 25% value range)
- framesize = number of preceding values to use for mean calculation (default = 1)
- iterations = number of flag-check iterations to perform to minimize inappropriate flagging of in-bounds values following out-of-bounds values (default = 1, max = 10; each iteration removes the first flagged value in each repetitively flagged group and re-analyzes the remaining values)

outputs:

Iflag = logical index of values outside of the range: mean-lowlimit < value > mean+highlimit

(c)2002-2013 Wade M. Sheldon and the Georgia Coastal Ecosystems LTER Project-2005 Wade M. Sheldon and the Georgia Coastal Ecosystems LTER Project

This file is part of the GCE Data Toolbox for MATLAB(r) software library.

The GCE Data Toolbox is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Automatic Assignment (Rules)

► Basic Examples (MATLAB conditionals, built-in functions)

Q/C Goal	Rule Type	Example
Limit/range check	simple conditionals	$x < 0 = 'I'; x > 37 = 'Q'$
		$\text{col_Salinity} < 0 = 'I'; \text{col_Salinity} > 37 = 'Q'$
Valid value check	null check function	$\text{isnull}(x) = 'M'$ (numeric or string column)
		$\text{isnan}(x) = 'M'$ (numeric only)
Sanity/consistency check	algebraic equations	$\text{col_WetWeight} \leq \text{col_DryWeight} = 'Q'$
		$(\text{col_SpartinaPct} + \text{col_JuncusPct} + \text{col_BorrighiaPct}) > 100 = 'I'$
		$(\text{col_NO2} + \text{col_NO3}) > \text{col_NOX} = 'I'$
Outlier detection	statistical tests	$x > \text{mean}(x) + 3 * \text{std}(x) = 'Q'$
		$x > \text{mean}(\text{no_nan}(x)) + 3 * \text{std}(\text{no_nan}(x)) = 'Q'$ (NaN-safe)
Condition check	multi-column rules	$\text{col_Depth} \leq 0.2 = 'I'; \text{col_BatteryVolts} \leq 9 = 'Q'$ (in Salinity)

Automatic Assignment (Rules)

► More Complex Examples (custom flag functions)

Q/C Goal	Rule Type	Example
Code check	set-based function	flag_notinlist(col_Site,'GCE1,GCE2,GCE3,GCE4')='I'
		flag_notinlist(col_Site, 'plant_list.mat\data\Species')='I'
		flag_notinarray(col_Plot,[1 5 10 15 20])='I'
Pattern check	moving window function	flag_valuechange(col_AirTemp,5,5,3)='Q'
		flag_nsigma(col_Humidity,3,3,10)='Q'
Location check	geographic lookup fnc	flag_locationcoords(x,'Longitude','Latitude',0.5)='Q'
Stuck/fouled sensor	inverse change function	flag_novaluechange(col_Salinity,0.2,0.2,3)='Q'
Derived property	custom function	flag_o2saturation(col_O2Conc,col_WaterTemp,col_Salinity,100,30,'mg/L')='Q'
Conditional checks	compound rules	flag_valuechange(col_AirTemp,5,5,3)&col_Precip==0='Q'
		col_PAR>0&flag_timeofday(col_Date,{ '00:00', '06:00', '20:00', '23:59:59' })='Q'

Automatic Assignment (Rules)

- ▶ Rules are evaluated automatically during processing
 - When data are imported and metadata template is applied
 - When data values are revised or columns added
 - When Q/C criteria edited or new template applied
- ▶ When rules are evaluated, any existing flags are overwritten
- ▶ Rules with syntax errors will be skipped, warning issued
- ▶ Rules can be “locked” & “unlocked” to control automatic assignment
 - Adding the token “manual” to the Q/C criteria locks flags and prevents recalculation
 - Rules are retained, but dataflag() will skip evaluation and never clear flags
 - Removing the token “manual” unlocks flags and triggers recalculation
- ▶ Rules are converted to column descriptions when flags are converted to text columns for export

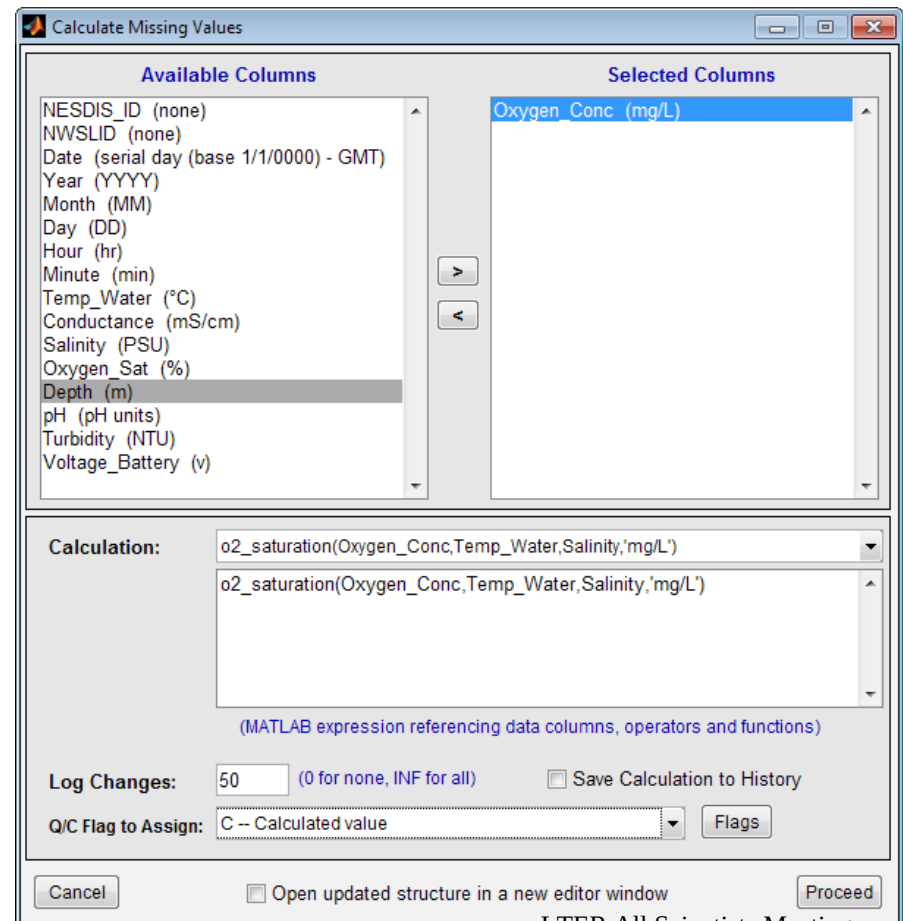
Automatic Assignment (Rules)

- ▶ Missing (NaN) values complicate rule evaluation
 - NaN can never equal anything, even another NaN (like RDBMS Null)
 - Aggregate statistics (mean, min, max, ...) return NaN if **any values** are missing
 - Compound Q/C criteria will always return false if any value is NaN, e.g. `col_Salinity>0&col_Depth>1&col_Temp>10&col_Temp<35='Q'` will never assign a flag for a row where Salinity, Depth or Temp value = NaN
 - No warning will be produced if NaN prevents flag assignment (valid condition)
- ▶ Best practices for NaNs
 - Always include basic, single-variable rules for limit checks when applicable (don't rely on multi-column rules)
 - Use `no_nan()` to remove NaNs from data before calculating mean, min, max, ...
 - Use `nan2zero()` to convert NaNs to 0 when rules involve totals

```
>> x = [2.1,7.25,1.2,-1,NaN,2.3]
x =
    2.1000    7.2500    1.2000   -1.0000    NaN    2.3000
>> idx = x == NaN
idx =
    0     0     0     0     0     0
>> idx = isnan(x)
idx =
    0     0     0     0     0     1
>> mean(x)
ans =
    NaN
>> mean(no_nan(x))
ans =
    2.3700
```

Flags for Revised/Calc'd Values

- ▶ Functions that calculate or revise values include options for assigning flags
 - Interpolated values – `interp_missing()`
 - Calculated values – `add_calcexpr()`, `calc_missing()`
 - Revised values – `update_data()`
- ▶ Useful for tracking data provenance and revisions



Flags for Summarized Data

- ▶ Functions that aggregate or bin data include options for generating Q/C criteria to flag derived columns
- ▶ Relevant metrics columns included automatically
 - Num_Missing_xxx
 - Pct_Missing_xxx
- ▶ Preserves info about quality and completeness of primary data
- ▶ Rules and flag assignments can be edited in derived data set using same tools as primary data set

Date/Time Interval Statistics

File Edit

Date/Time Interval for Statistical Summaries: Daily

Available Columns

- Provisional (none)
- Depth (m)
- Water_Level (m)
- Precipitation (mm)
- Conductance (microsiemens/cm at 25°C)
- Oxygen_Sat (%)
- pH (pH units)
- Turbidity (FNU)
- WindSpeed (m/s)
- WindDirection (degrees)
- Temp_Air (°C)
- Baro_Pressure (mbar)

Column Selections

Group Records By (optional)

- Agency (none)
- StationID (none)

Calculate Statistics For

- Temp_Water (°C)
- Salinity (mg/ml)
- Oxygen (mg/L)

☐ Remove values flagged <any> List

Flag as I if percent missing > 30

Flag as Q if consecutive mis... 4

Flag as I if percent flagged > 40

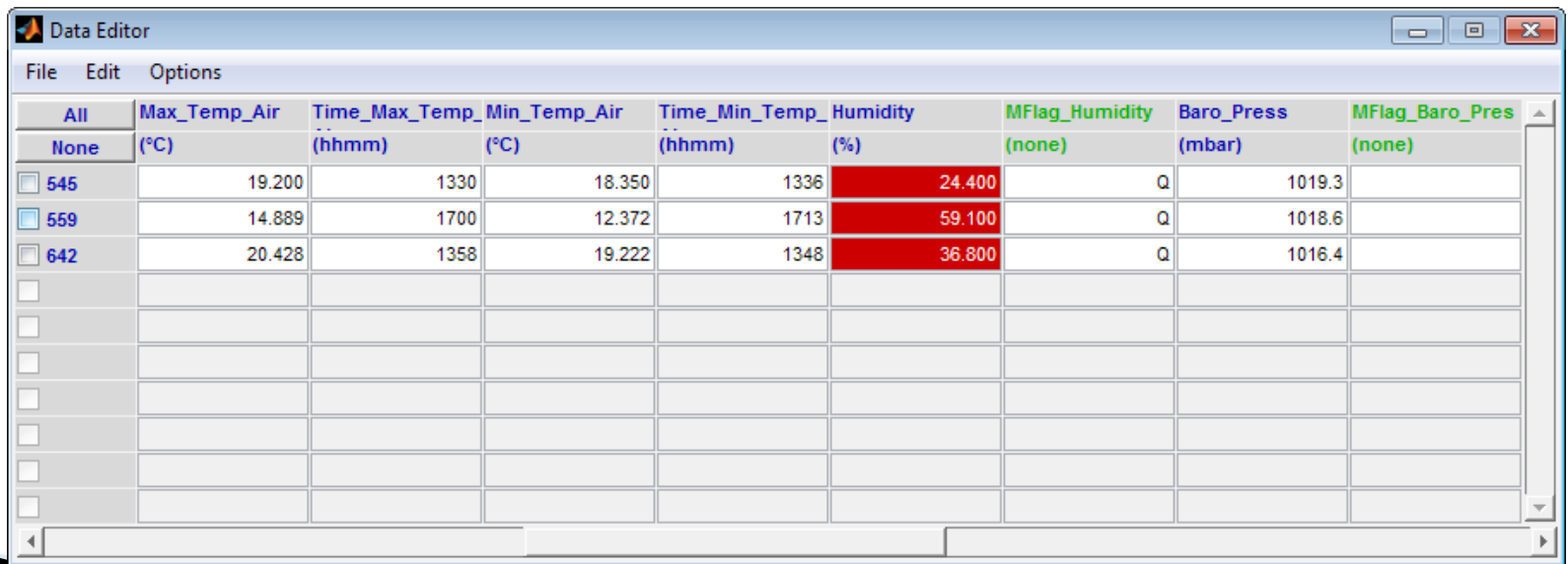
Flag as Q if percent flagged > 10

Statistics: Automatic (based on numeric and variable type) ☐ Time Min/Max

Cancel ☒ Close dialog after exporting LTER All Scientists Meeting Proceed

Manual Assignment (Grid)

- ▶ Flags can be manually edited in the data grid
 - Useful for fine-tuning assignments
 - Can add and remove flags
 - Can filter view to hone in on flags or values
 - Any edits “lock” rules to prevent overwriting revisions

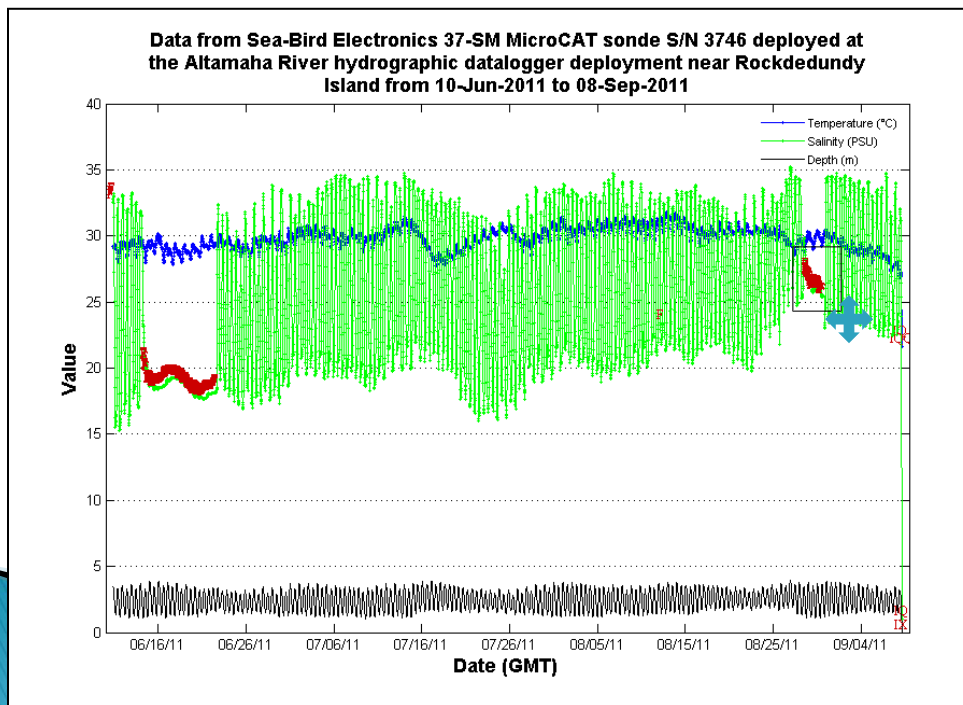


The screenshot shows a window titled "Data Editor" with a menu bar (File, Edit, Options) and a data grid. The grid has columns for various environmental variables and their flags. The first column contains checkboxes and IDs (545, 559, 642). The second column is Max_Temp_Air (°C). The third column is Time_Max_Temp (hhmm). The fourth column is Min_Temp_Air (°C). The fifth column is Time_Min_Temp (hhmm). The sixth column is Humidity (%). The seventh column is MFlag_Humidity (none). The eighth column is Baro_Press (mbar). The ninth column is MFlag_Baro_Pres (none). The data rows show values for three specific data points: 545, 559, and 642. The humidity values (24.400, 59.100, 36.800) are highlighted in red.

	All	Max_Temp_Air	Time_Max_Temp	Min_Temp_Air	Time_Min_Temp	Humidity	MFlag_Humidity	Baro_Press	MFlag_Baro_Pres
	None	(°C)	(hhmm)	(°C)	(hhmm)	(%)	(none)	(mbar)	(none)
☒	545	19.200	1330	18.350	1336	24.400	Q	1019.3	
☒	559	14.889	1700	12.372	1713	59.100	Q	1018.6	
☒	642	20.428	1358	19.222	1348	36.800	Q	1016.4	
☐									
☐									
☐									
☐									
☐									
☐									
☐									
☐									

Graphical Assignment

- ▶ Visual Q/C tool can be invoked from interactive data plots
 - Actions variable-specific to prevent inadvertent flagging of wrong values
 - Can zoom/pan to isolate points to flag
 - Left-click/drag to assign, right-click/drag to clear
- ▶ Anomaly reports can be auto-generated and edited to describe rationale in metadata



Visual QC/QA Tool

Data Column to Flag: Salinity

Flag Code to Assign: Q = questionable value

=

(Note: Left click/drag to assign flags, right click/drag to clear flags)

Data Anomalies Metadata

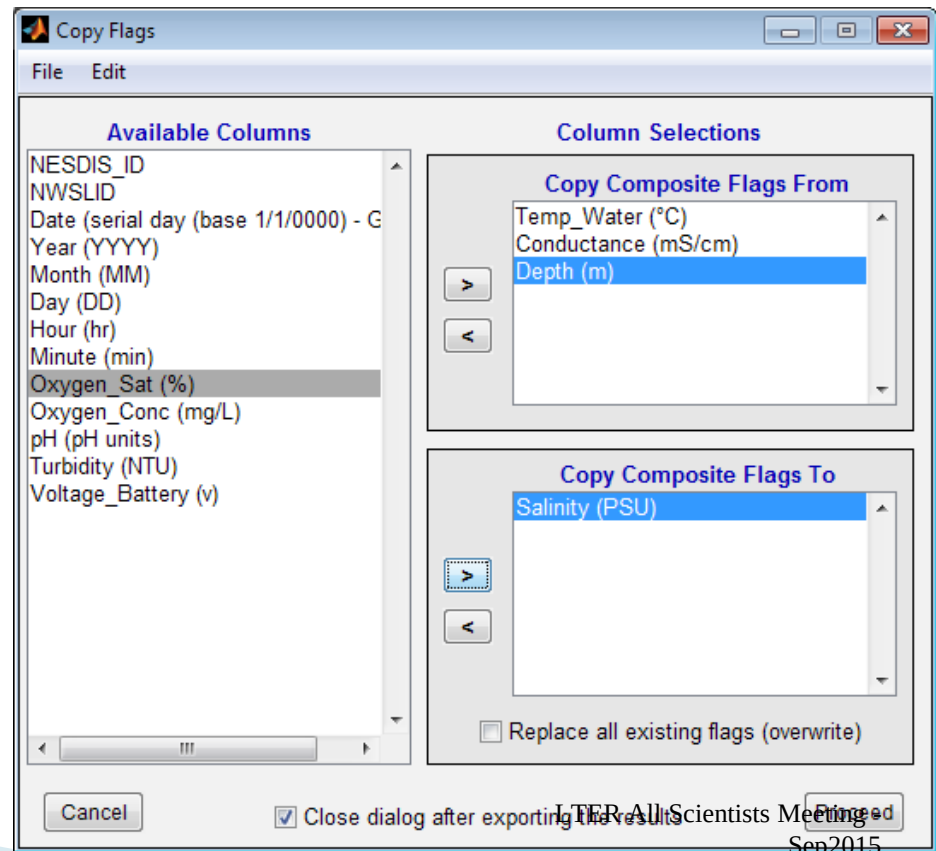
☒ Include Missing **Format** mm/dd/yyyy

Values of 2 records in Temperature were flagged as I (invalid value (out of range)) for date 09/08/2011. Values of 2 records in Conductivity were flagged as I (invalid value (out of range)) for date 09/08/2011. Values of 24 records in Conductivity were flagged as Q (questionable value) for dates: 06/13/2011, 06/22/2011, 07/04/2011, 07/05/2011, 07/06/2011, 07/07/2011, 07/10/2011, 07/13/2011, 07/14/2011, 07/30/2011, 07/31/2011, 08/01/2011, 08/02/2011, 08/04/2011, 08/05/2011, 08/11/2011, 08/12/2011, 08/15/2011. Values of 2 records in Pressure were flagged as I (invalid value (out of range)) for date 09/08/2011. Values of 2 records in Depth were flagged as I (invalid value (out of range)) for date 09/08/2011. Values of 469 records in Salinity were flagged as F (undefined flag) for dates: 06/10/2011, 06/14/2011, 06/15/2011-06/22/2011,

☒ Automatically Close

Copying Flags

- ▶ Flags can be copied from 1 + columns to 1 + other columns (many:many cardinality)
- ▶ Useful for propagating flags to dependent columns
 - After manual editing
 - To avoid complex dependency checks
- ▶ Can add to or over-write flags on target column(s)
- ▶ Can use GUI form or `copyflags()` function from command line



Text Columns to Flags

- ▶ Flags imported as text columns can be converted to intrinsic toolbox flags
 - Single and multi-character flags supported
 - Multi-character flags require mapping (/settings/flap_mapping.mat)
- ▶ Flag columns must have “Flag_” prefix and name matching corresponding data column (Salinity, Flag_Salinity)
- ▶ Flags can be added to or overwrite existing flags
- ▶ Can be used to cache manual flag assignments before restoring automatic evaluation
 - Instantiate flags as columns
 - Edit manual flags as necessary
 - Unlock flags and revise Q/C criteria as necessary
 - Convert text columns to flags, adding to new automatically-assigned flags

Specialized Flag Assignment Func's

- ▶ `daterange2flags()`
 - Assigns flag to 1 or more columns based on date ranges
 - Accepts array of start/end dates (discontinuous)
 - Useful for flagging known outages or maintenance periods
 - Command-line only
- ▶ `codes2criteria()`
 - Generates Q/C criteria for coded columns based on code definitions in the metadata
 - Useful for catching typos and undefined codes
 - Command-line or Data Editor menu command

Managing Qualifiers

- ▶ Qualifiers can be cleared using the same approaches
 - Revise Q/C criteria limits or remove rules
 - Clear flags manually or graphically on plots (right-click)
 - Copy flags from other columns using the overwrite option
- ▶ Criteria can be locked/unlocked to manage auto assignment
 - Locking adds the token “manual” to criteria (but preserves other rules)
 - Unlocking removes “manual”, triggers recalculation
 - Any manually-assigned flags cleared when criteria unlocked!
- ▶ Bulk operations also supported
 - Clear all flags in selected column(s) and lock rules
 - Revise flag codes using search/replace



Managing Qualified Values

- ▶ Data values and qualifier flags paired and managed together transparently
 - Flags move with data when sorting rows/columns, merging, joining data
 - Flag column present even if no criteria defined or flags are assigned
 - Revising data value triggers criteria evaluation (unless locked)
- ▶ Data never automatically altered/deleted based on flags
 - Flags regarded as value-level metadata
 - Separation of data, flags permits flexible handling in workflows, archives
- ▶ Data analysis, plotting, export tools provide options for managing flagged values “on the fly”
 - Retain, document/display flags with values
 - Delete all or selected flagged values
 - Ignore flags

Managing Qualified Values

- ▶ Can document flagged/missing values in metadata
- ▶ Can selectively delete flagged values, rows permanently

Edit Flag Definitions/Anomalies

Quality Control Flag Code Definitions

I = invalid data value
Q = questionable data value
e = value estimated by USGS
P = provisional data value
1 = value locked in the USGS database for unspecified reason
E = data value estimated by interpolation

Move First
Move Up
Move Down
Move Last
Add New
Delete

Definition: I = invalid data value Update

Data Anomalies Report Clear

Values of 958 records in Discharge are missing for dates: 08/25/2014-09/04/2014. Values of 958 records in GageHeight are missing for dates: 08/25/2014-09/04/2014. Values of 6 records in Temp_Water were flagged as E (data value estimated by interpolation) for dates: 11/05/2014, 11/20/2014, 12/01/2014. Values of 964 records in Discharge were flagged as E (data value estimated by interpolation) for dates: 08/25/2014-09/04/2014, 11/05/2014, 11/20/2014, 12/01/2014. Values of 964 records in GageHeight were flagged as E (data value estimated by interpolation) for dates: 08/25/2014-09/04/2014, 11/05/2014, 11/20/2014, 12/01/2014. Values of 6 records in Precipitation are missing for dates: 11/05/2014, 11/20/2014, 12/01/2014.

Cancel Accept

Flagged Value Deletion

Available Columns

Agency (none)
StationID (none)
Provisional (none)
Date (serial day (base 1/1/0000) - GMT)
Year (YYYY)
Month (MM)
Day (DD)
Hour (hr)
Minute (min)

Selected Columns

Temp_Water (°C)
Discharge (m³/sec)
GageHeight (m)
Precipitation (mm)

Flagged Values to Retain

Q = questionable data value
e = value estimated by USGS
P = provisional data value
1 = value locked in the USGS database
E = data value estimated by interpolation

Flagged Values to Delete

I = invalid data value

Value Deletion Option: ☒ Convert to NaN/blank ☐ Delete Entire Row

Flag Deletion Option: ☐ Clear flags ☒ Retain/lock flags ☐ Replace/lock flags

New Flag: < no flag assigned > Flags

Other Options: ☒ Log deletions to data anomalies metadata field

Cancel Proceed

Q/C Strategy Recommendations

- ▶ Define limit check rules for all measured variables
 - Sensor reading outside max/min specs
 - Impossible values (e.g. negative count, fraction over 100%, zero mass)
 - Improbable values based on historical data, experience (e.g. >3 SD over mean)
- ▶ Define set-based rules to check coded values (flag_inlist, ...)
- ▶ Define multi-column dependency rules for related variables
 - Redundant measurements comparable (e.g. air temp)
 - Suitable conditions when measurements made (e.g. good water depth, above freezing, suitable wind speed)
 - Reasonable time of day for diurnal variables (e.g. PAR, solar rad day/night)
 - Readings not recorded during extreme events (e.g. precip, wind)
- ▶ Define rules to check for expected disruptions or maintenance
 - Sensors disconnected or out of water for cleaning
 - Stuck/fouled/blocked sensors

Q/C Strategy Recommendations

- ▶ Define pattern check rules to identify possible contamination
 - Value changes too rapid (flag_valuechange, flag_nsigma)
 - Too little value change over time (flag_novaluechange)
 - Short-duration spikes/dips (flag_valuechange, flag_percentchange)
 - Long-duration maintenance events (flag_well_pumping)
- ▶ Save baseline Q/C rules in metadata template for re-use
- ▶ Review/refine automatic flag assignments on data plots
- ▶ Propagate flags to dependent/calculated columns to reflect edits on primary measurements
- ▶ Generate anomalies report and edit to include descriptions of rationale for manual revisions

Demo/Exercises

- ▶ Import or load a sample data set
- ▶ Define flags and rules in Q/C Criteria editor
 - Add and define one or more flags using 'Flag' button
 - Define limit check rules using primary form controls
 - Define multi-column rules using 'Add/Edit Custom Criteria'
 - Define algorithmic rules using 'Add/Edit QC Flag Function'
 - Try to implement all of the syntax examples from slides 9–10
 - Save the metadata as a template for re-use
- ▶ Refine flags on a data plot
 - Add single flags and drag to flag a range of values
 - Clear flags
- ▶ Copy flags from multiple columns to another column
- ▶ Unlock flags to restore automatic calculation
- ▶ Delete values assigned a specific flag
- ▶ Interpolate missing values and assign a flag to the imputed values

Core QA/QC Functions

<code>addflags</code>	Adds a character flag to specified records in one or more columns of a GCE Data Structure syntax: <code>[s2, msg] = addflags(s, cols, rows, flag, flagdef)</code>
<code>clearflags</code>	Clears specified flags from a GCE Data Structure for display or export purposes syntax: <code>[s2, msg] = clearflags(s, flagchars, cols, lockopt, metaopt)</code>
<code>codes2criteria</code>	Generates QA/QC criteria for coded columns in a GCE Data Structure based on code definitions syntax: <code>[s2, msg] = codes2criteria(s, cols, flagcode)</code>
<code>cols2flags</code>	Converts values in specified text columns of a GCE Data Structure to QA/QC flags syntax: <code>[s2, msg] = cols2flags(s, flagcols, datacols, overwrite, delete, prefix)</code>
<code>cols2flags_mapped</code>	Converts data set columns to QA/QC flags after mapping multi-character flags to single character syntax: <code>[s2, msg] = cols2flags_mapped(s, flagset_name, flagcols, datacols, overwrite, delete)</code>
<code>copyflags</code>	Copies composite flags from one or more columns and adds to or replaces the existing flag arrays syntax: <code>[s2, msg] = copyflags(s, flagcols, datacols, opt)</code>
<code>cullflags</code>	Deletes all records from a GCE Data Structure containing any values assigned specified flags syntax: <code>[s2, msg] = cullflags(s, flagchars, cols, metaopt, logsize)</code>
<code>dataflag</code>	Evaluates Q/C criteria or assigns user-specified flags to generate or update Q/C flag arrays syntax: <code>[s2, msg, flagerrors] = dataflag(s, cols, manual_flags)</code>
<code>daterange2flags</code>	Assigns Q/C flags by date range and locks flags to prevent automatic recalculation syntax: <code>[s2, msg] = daterange2flags(s, datecol, dates, cols, flag)</code>
<code>get_flagdefs</code>	Retrieves definitions for selected QA/QC flags from GCE Data Structure metadata syntax: <code>flagdefs = get_flagdefs(s, flags)</code>
<code>harvest_check</code>	Generates a harvest check email based on user-specified thresholds for missing and flagged values syntax: <code>msg = harvest_check(s, days, missing, flagged, daily_flagged, wrap)</code>
<code>nullflags</code>	Converts values in a GCE Data Structure assigned specified flags to NaN/empty syntax: <code>[s2, msg] = nullflags(s, flagchars, cols, metaopt, clearflags, newflag, logsize)</code>

QA/QC GUI Applications

<code>ui_clearflags</code>	GCE Data Toolbox dialog for selective deletion of QA/QC flags or flagged values, or selective conversion of flags to columns syntax: <code>ui_clearflags(op, s, mode, h_cb, cb)</code>
<code>ui_copyflags</code>	GCE Data Toolbox dialog for copying composite flags from one or more data columns syntax: <code>ui_copyflags(op, s, h_cb, cb)</code>
<code>ui_flagdefs</code>	QA/QC flag definition and data anomalies editor dialog called by 'ui_editor'. syntax: <code>ui_flagdefs(op, meta, h_cb, cb)</code>
<code>ui_flagfunction</code>	GUI dialog called by 'ui_qccriteria' for adding custom function calls to a Q/C criteria string syntax: <code>ui_flagfunction</code>
<code>ui_flagpicker</code>	Adds a Q/C flag picker popupmenu and edit button to a GUI figure at a specified position syntax: <code>ui_flagpicker(op, s, pos, h, cb, enable)</code>
<code>ui_manual_qc</code>	Opens a GCE Data Structure in a data grid to allow data values and Q/C flags to be viewed and edited syntax: <code>ui_manual_qc(op, s, cols, h, cb, colwid)</code>
<code>ui_qccriteria</code>	QA/QC criteria editor dialog called by 'ui_editor' and 'ui_template' syntax: <code>ui_qccriteria(op, s, col, flagdefs, h_cb, cb)</code>
<code>ui_querybuilder</code>	GUI dialog for building row restriction queries to subselect data from a GCE Data Structure syntax: <code>ui_querybuilder(op, val)</code>
<code>ui_string_replace</code>	Dialog called by 'ui_editor' for searching and replace text values or flags in a GCE Data Structure syntax: <code>ui_string_replace(op, s, col, target, h_cb, cb)</code>
<code>ui_visualqc</code>	Dialog for assigning and clearing QC/QA flags visually by clicking on data points with the mouse. syntax: <code>ui_visualqc(op, qc_data)</code>

QA/QC Functions

<code>flag_locks</code>	Locks or unlocks Q/C flag criteria for specified columns in a GCE-LTER data structure syntax: <code>[s2, msg] = flag_locks(s, lock_option, cols)</code>
<code>flag_replace</code>	Performs string replacement for flags of one or more columns in a GCE Data Structure syntax: <code>[s2, msg] = flag_replace(s, cols, oldstring, newstring, matchtype, caseopt)</code>
<code>flags2cols</code>	Converts Q/C flags in a GCE Data Structure to coded string or integer columns in the data set syntax: <code>[s2, msg] = flags2cols(s, flagcols, clearopt, missing, pos, encode, prefix, firstflag)</code>
<code>flags2cols_selected</code>	Converts selected Q/C flag information in specified columns of a GCE Data Structure to coded string columns syntax: <code>[s2, msg] = flags2cols_selected(s, flags, cols, encode, emptycols, prefix)</code>
<code>flag_ctd_soak_period</code>	Returns an index indicating records collected during the pre-deployment soaking period of a CTD cast syntax: <code>Iflag = flag_ctd_soak_period(vals, windowsize, maxdepth)</code>
<code>flag_daterange</code>	Returns an index of date/time values that are within a specified time of day for time-based QA/QC flagging syntax: <code>Iflag = flag_daterange(datetime, daterange, dateformat)</code>
<code>flag_inarray</code>	Returns an index of numeric values that are present in a specified array syntax: <code>Iflag = flag_inarray(vals, valuearray, tolerance)</code>
<code>flag_inlist</code>	Returns an index of string values that are present in a specified list or file syntax: <code>Iflag = flag_inlist(vals, valuelist, caseopt)</code>
<code>flag_locationcoords</code>	Returns an index of location codes with coordinates that differ by more the specified tolerance from reference coordinates syntax: <code>Iflag = flag_locationcoords(location, longitude, latitude, tolerance, caseoption, ref_file)</code>
<code>flag_locationnames</code>	Returns an index of location code values that are not present in the geographic database syntax: <code>Iflag = flag_locationnames(locations, caseopt, emptyopt)</code>

QA/QC Functions (cont.)

<code>flag_notinarray</code>	Returns an index of numeric values that are not present in a specified array syntax: <code>Iflag = flag_notinarray(vals, valuearray, tolerance)</code>
<code>flag_notinlist</code>	Returns an index of string values that are not present in a specified list or file syntax: <code>Iflag = flag_notinlist(vals, valuelist, caseopt, emptyopt)</code>
<code>flag_novaluechange</code>	Returns an index of values that do not differ from the mean of preceeding values syntax: <code>Iflag = flag_novaluechange(vals, lowlimit, highlimit, framesize, iterations)</code>
<code>flag_nsigma</code>	Returns an index of values above or below the mean of preceeding values by the specified number of standard deviations syntax: <code>Iflag = flag_nsigma(vals, lowlimit, highlimit, framesize, iterations)</code>
<code>flag_o2saturation</code>	Returns an index of DO concentrations that are above or below specified saturation limits syntax: <code>Iflag = flag_o2saturation(o2_conc, temp, sal, maxsat, minsat, units)</code>
<code>flag_percentchange</code>	Returns an index of values above/below the mean of preceeding values by specified percentage syntax: <code>Iflag = flag_percentchange(vals, lowlimit, highlimit, framesize, iterations)</code>
<code>flag_sitenames</code>	Returns an index of site code values that are not present in the geographic database syntax: <code>Iflag = flag_sitenames(sitecodes, caseopt, emptyopt)</code>
<code>flag_timeofday</code>	Returns an index of date/time values that are within a specified time of day syntax: <code>Iflag = flag_timeofday(datetime, timerange, dateformat)</code>
<code>flag_total</code>	Returns an index of values that exceed a limit when totalled with a specified number of preceding values syntax: <code>Iflag = flag_total(vals, highlimit, framesize, iterations)</code>
<code>flag_valuechange</code>	Returns an index of values above or below the mean of preceeding values by a limit syntax: <code>Iflag = flag_valuechange(vals, lowlimit, highlimit, framesize, iterations)</code>
<code>flag_well_pumping</code>	Returns an index indicating records collected during and following well pumping events based on negative spikes in pressure syntax: <code>Iflag = flag_well_pumping(vals, windowsize, threshold)</code>