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

Wade Sheldon

Georgia Coastal Ecosystems LTER

University of Georgia

Introduction

- Quality Control of high volume, real-time data from automated sensors is an emerging challenge
  - Traditional techniques (plotting, stats) often don't scale well
  - Data validation and Q/C can be limiting factor in getting data "online"
  - Difficulties lead to release delays or posting provisional data
- GCE Data Toolbox (MATLAB-based software developed at GCE-LTER) has proven useful for Q/C of real-time data
- Designed to automate GCE data processing, QA/QC and metadata generation, but very generalized and supports any tabular data
- Provides **dynamic, rule-based Q/C framework** for data processing, analysis and synthesis

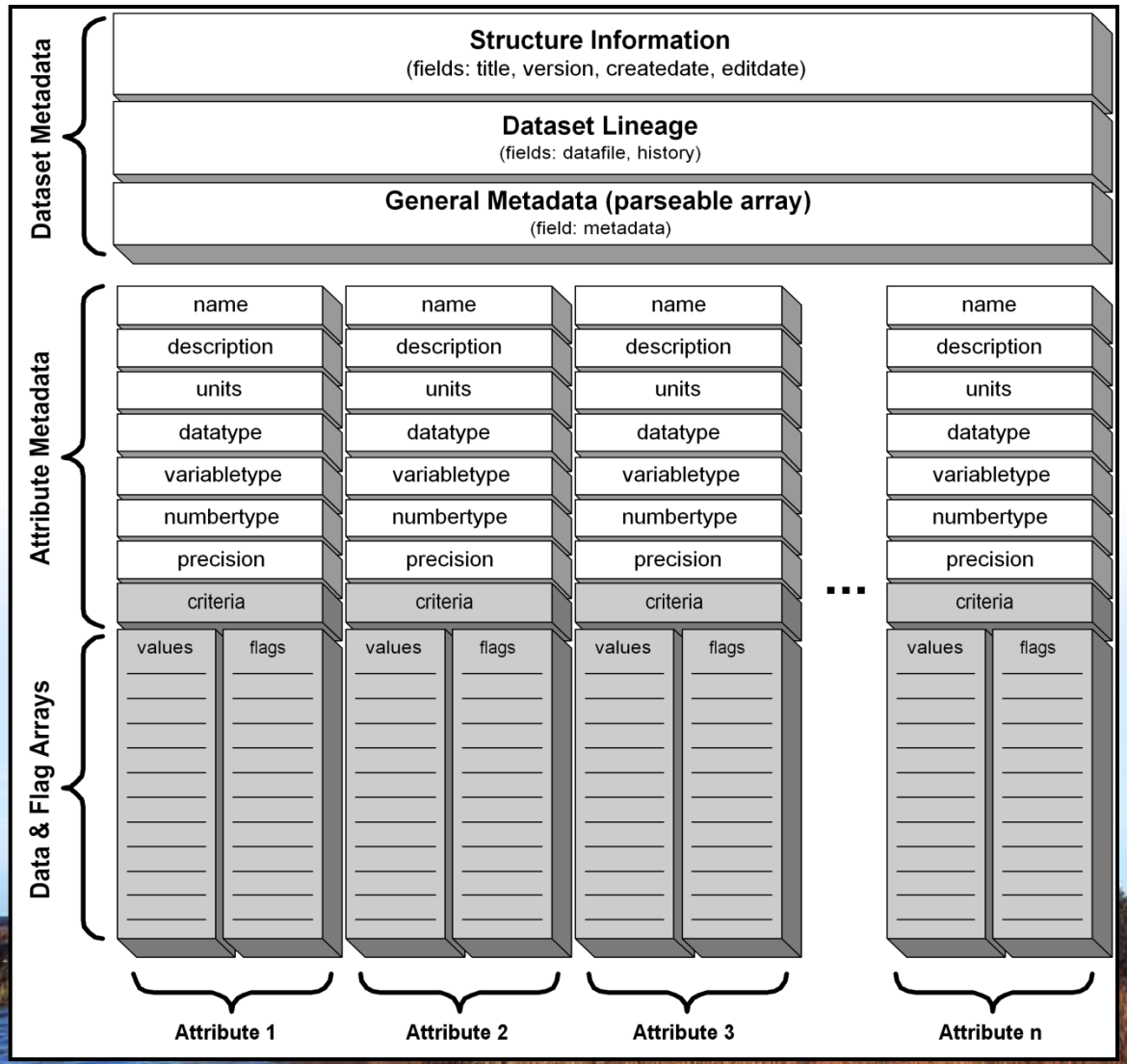
Q/C Framework Components

- Generalized tabular data model designed to support Q/C
- Software for Q/C analysis and qualifier flag management
- Software for Q/C-aware data analysis, synthesis

Q/C Framework Components

- Generalized tabular data model designed to support Q/C
  - Any size data table (numeric & text fields)
  - Detailed metadata (dataset-level documentation & attribute descriptors)
  - Q/C rules for every attribute, qualifier flags for every value
  - Data processing and Q/C operation history (lineage)

Data Model (GCE Data Structure)

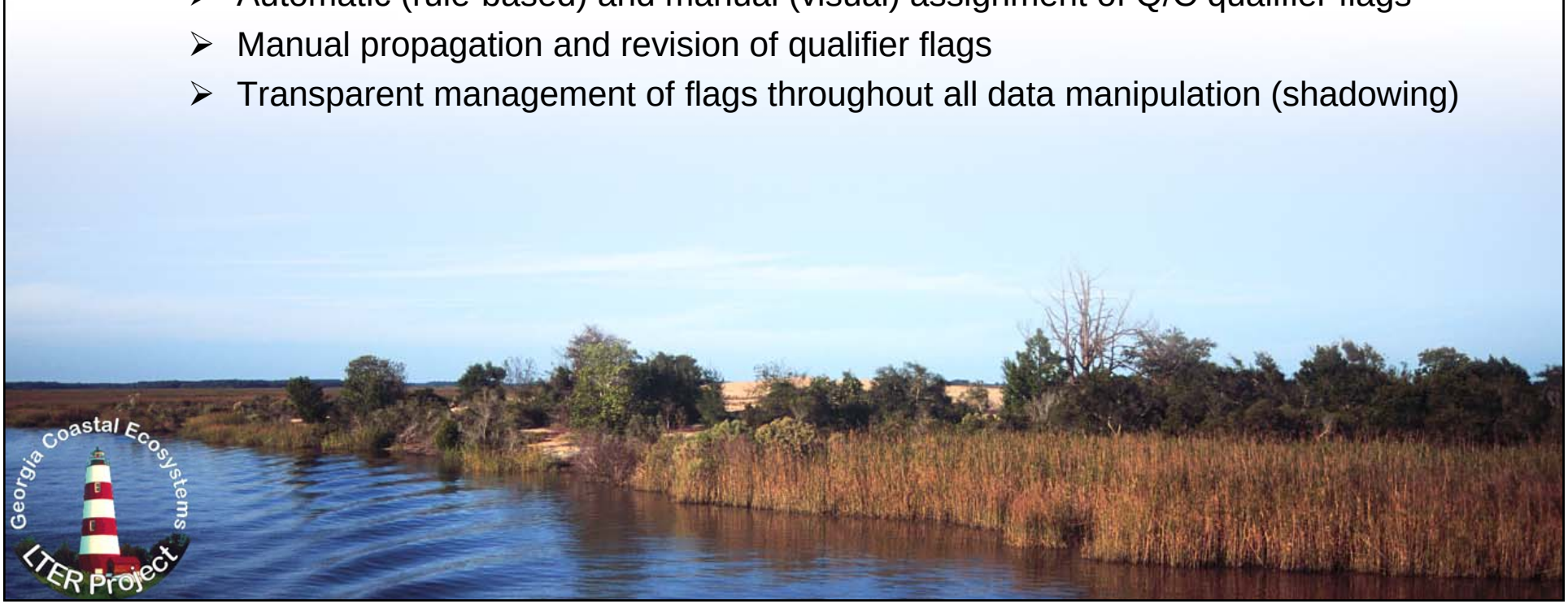


Georgia Coastal Ecosystems LTER Project

Q/C Framework Components

- Generalized tabular data model designed to support Q/C
  - Any size data table (numeric & text fields)
  - Detailed metadata (dataset-level documentation & attribute descriptors)
  - Q/C rules for every attribute, qualifier flags for every value
  - Data processing and Q/C operation history (lineage)

- Software for Q/C analysis and qualifier flag management
  - Automatic (rule-based) and manual (visual) assignment of Q/C qualifier flags
  - Manual propagation and revision of qualifier flags
  - Transparent management of flags throughout all data manipulation (shadowing)



Software for Q/C Analysis

- Programmatic/Algorithmic Q/C Analysis
  - Based on “rules” that define conditions in which values should be flagged
  - Unlimited Q/C rules can be defined for each attribute
  - Scope can range from single value to entire data set (+ external files, WS)
  - Rules evaluated when data loaded and when data or rules change
  - Rules can be predefined in metadata templates to automate Q/C on import

Q/C Rule Definitions

- Syntax: [logical expression]='[flag code]'; ...
  - [logical expression] defines conditions for which flags should be assigned (true/false)
  - [flag code] is alphanumeric character to assign when expression is true (I, q, 9, *)
  - Sophistication required about on par with equation writing in Excel

- Basic Examples

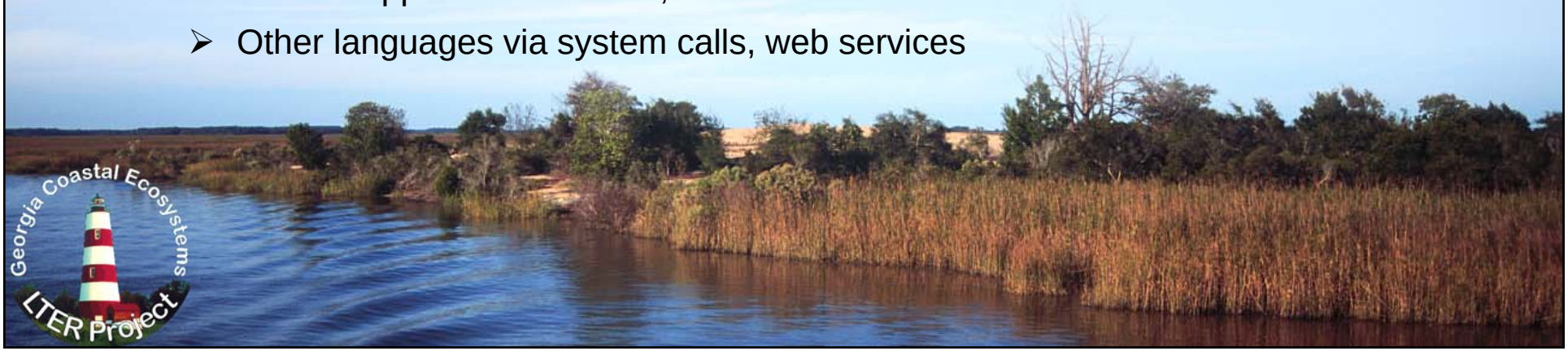
Q/C Goal	Rule Type	Example
Limit/range check	simple conditionals	col_Salinity<0='I';col_Salinity>37='Q'
		x<0='I';x>37='Q'
Sanity/consistency check	algebraic equations	(col_SpartinaPct+col_JuncusPct+col_BorrichiaPct)>100='I'
		(col_NO2+col_NO3)>col_NOX='I'
Outlier detection	statistical tests	x>mean(x)+3*std(x)='Q'
Condition check	multi-column rules	col_Depth<=0.2='I';col_BatteryVolts<=9='Q' (in Salinity)

Q/C Rule Definitions

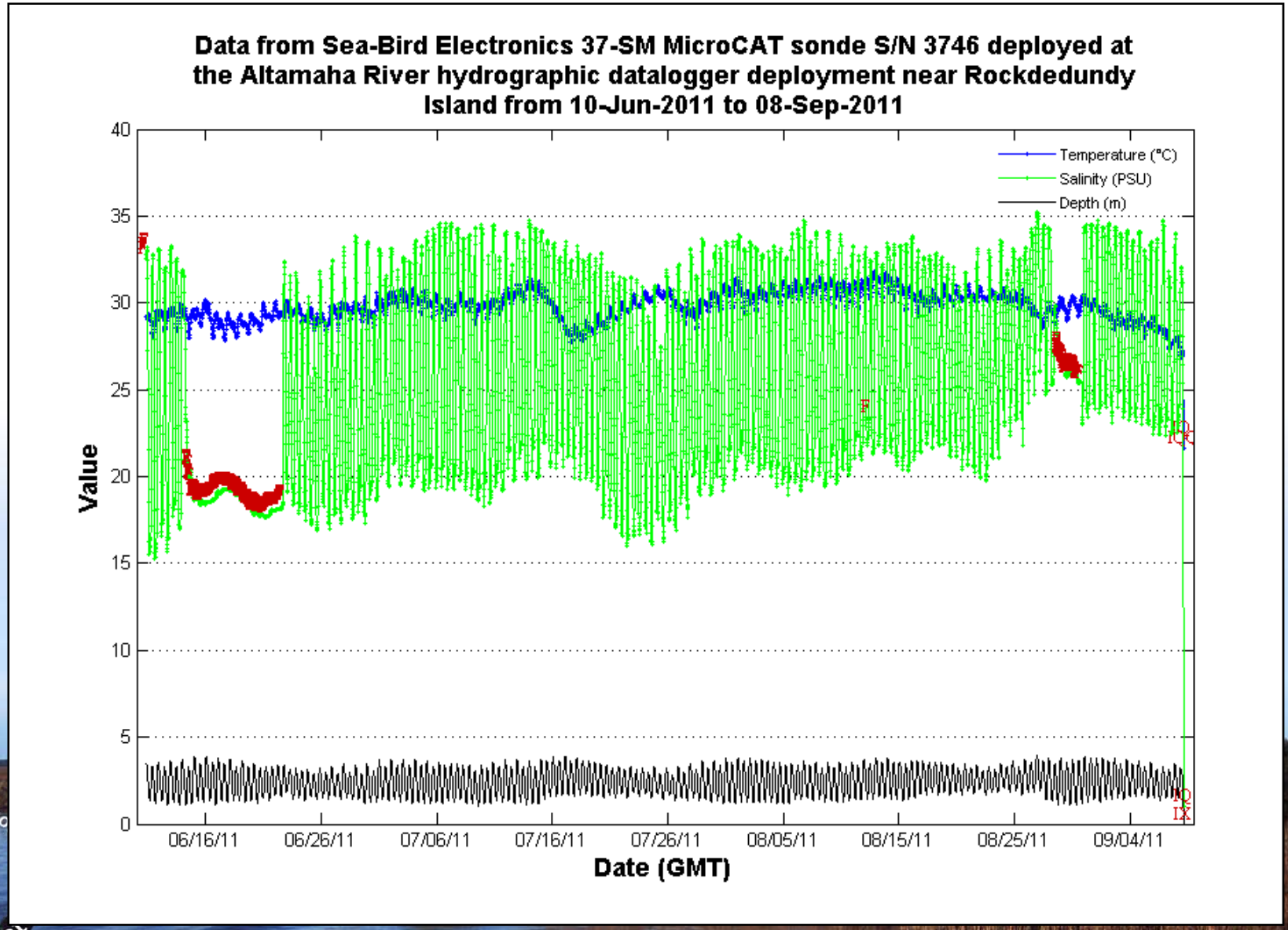
- More Complex Examples

Q/C Goal	Rule Type	Example
Code check	set-based function	flag_notinlist(col_Site,'GCE1,GCE2,GCE3,GCE4')='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'
Stuck/fouled sensor	inverse change function	~flag_valuechange(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'

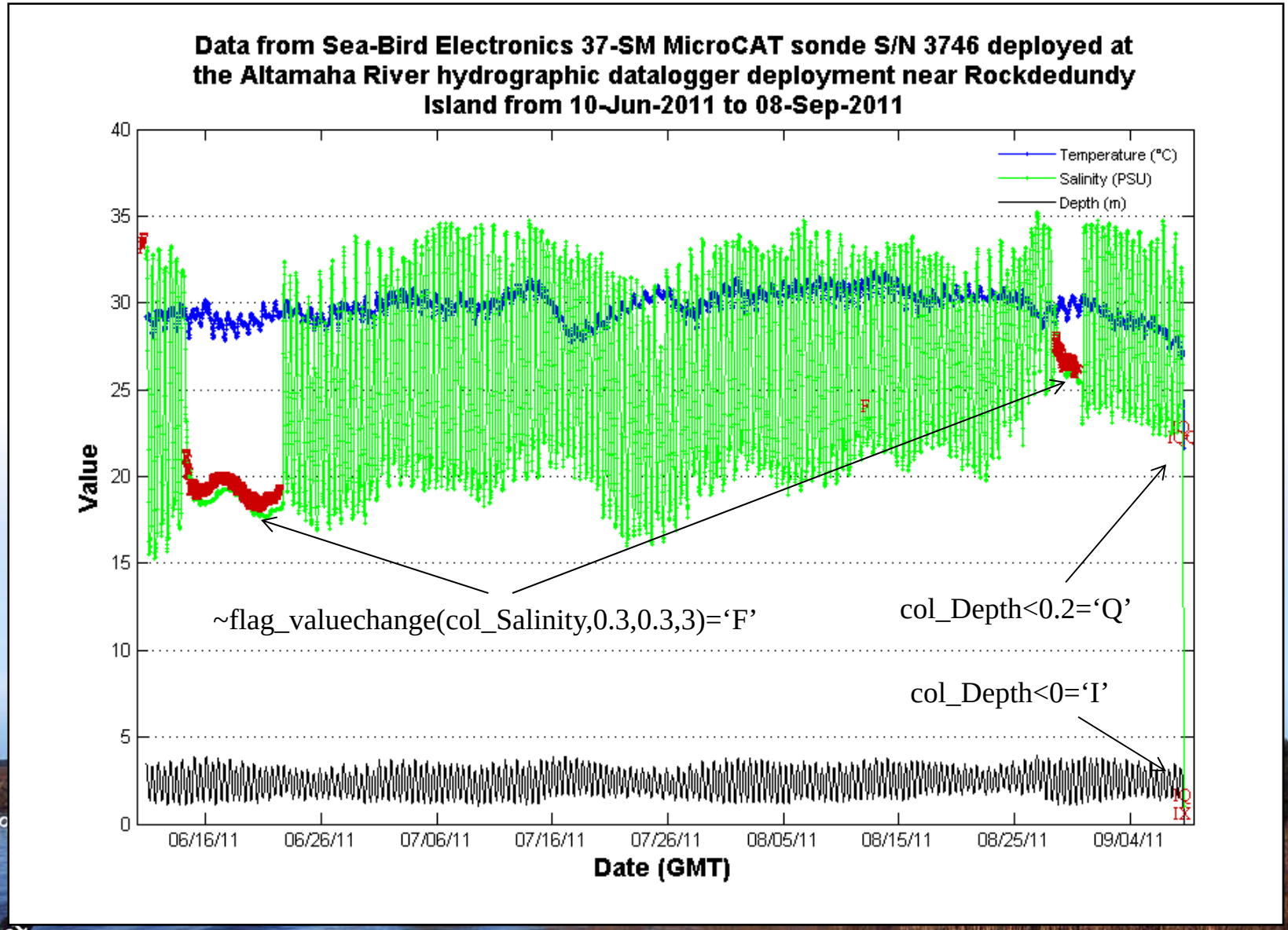
- Advanced computations/models can also be called in Q/C rules
  - Native support for MATLAB, Java code
  - Other languages via system calls, web services



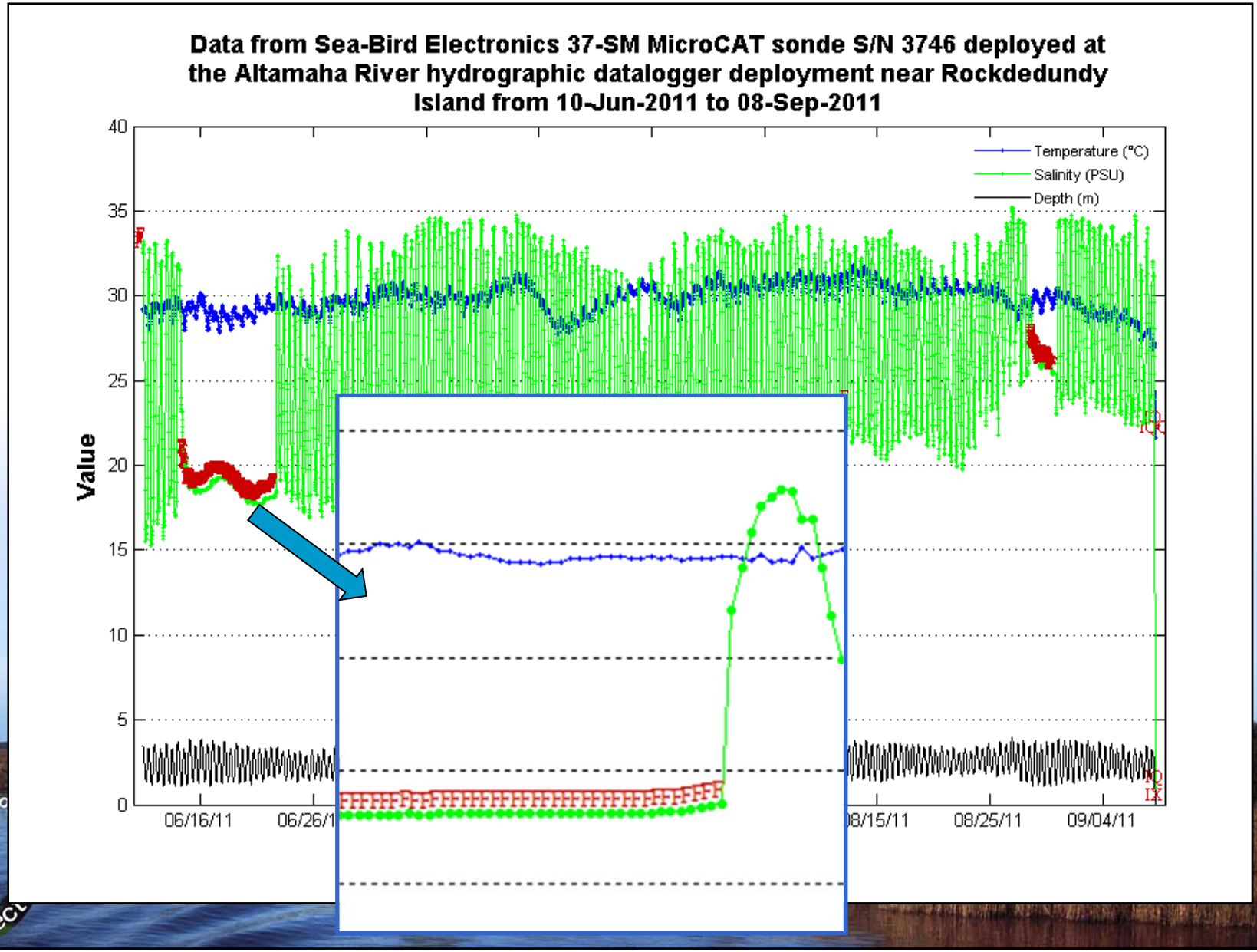
Example – CTD mooring



Example – CTD mooring

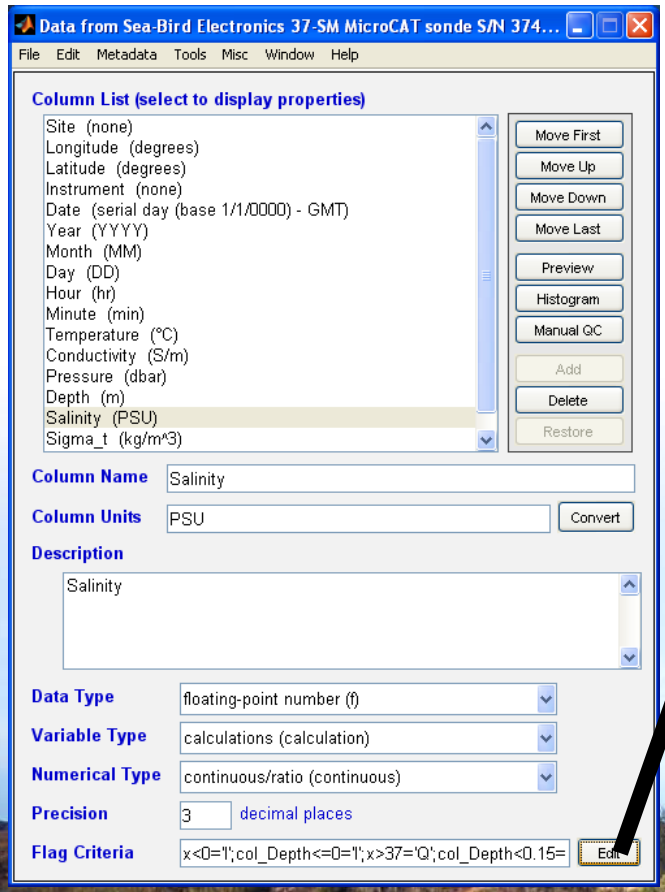


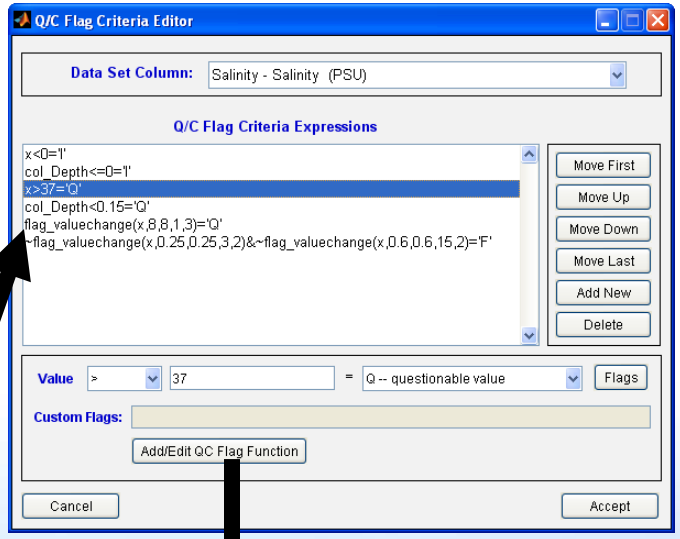
Example – CTD mooring

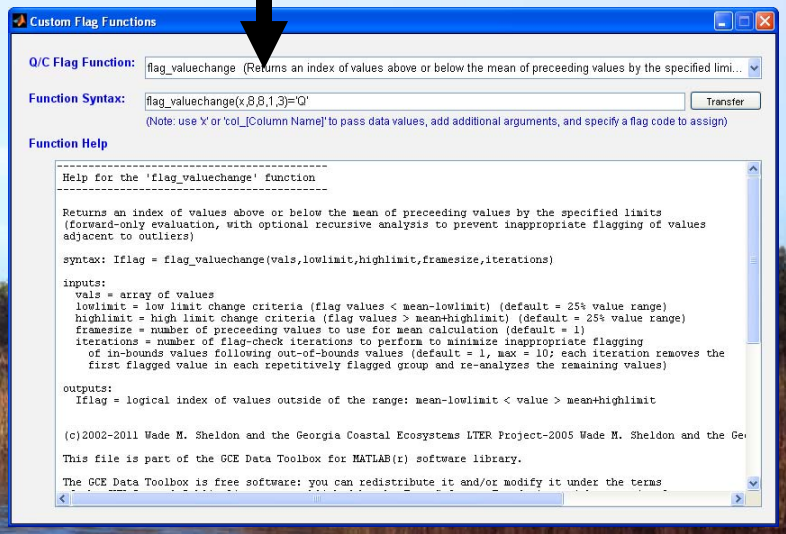


Q/C Rule Management

- Rules can be created, edited, deleted, re-ordered using GUI forms
- Syntax help is available for referencing parameterized Q/C functions





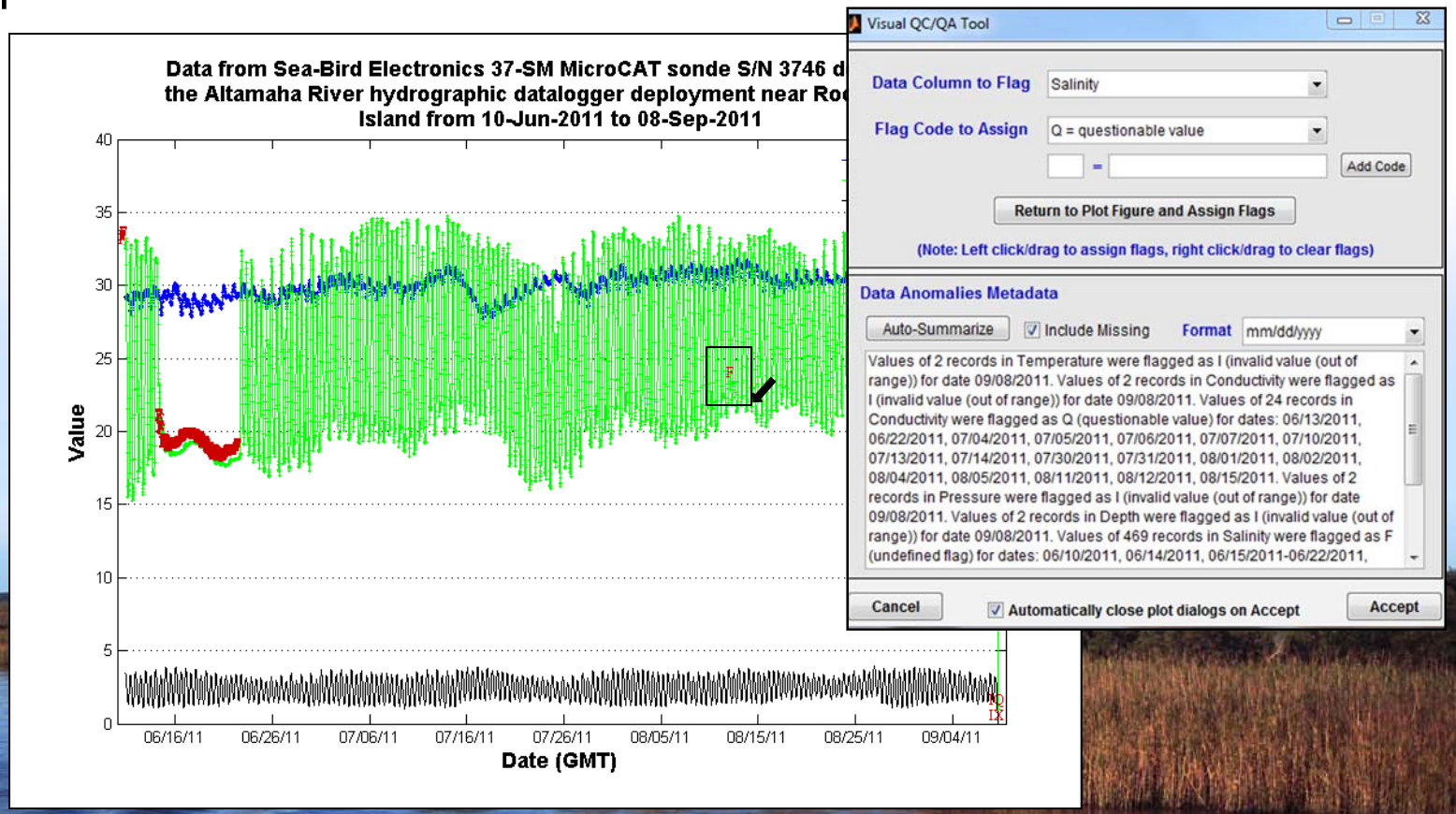


Software for Q/C Analysis

- Programmatic/Algorithmic Q/C Analysis
  - Based on “rules” – expressions evaluated to identify values matching criteria
  - Unlimited Q/C rules can be defined for each attribute
  - Scope can range from single value to entire data set (+ external files)
  - Rules evaluated when data loaded and when data or rules change
  - Rules can be predefined in metadata templates to automate Q/C on import
- Interactive Q/C Analysis and Revision
  - Qualifiers can be assigned/cleared visually on data plots with the mouse
  - Qualifiers can be propagated to dependent columns
  - Qualifiers can be removed or edited (search/replace) if standards change

Interactive Q/C Tools

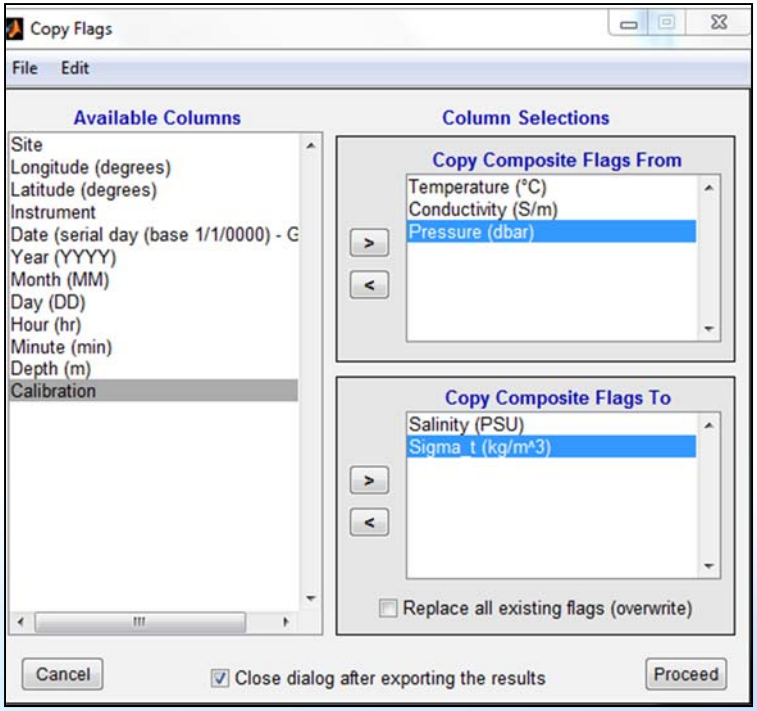
- Visual Q/C tool can be invoked from interactive data plots
  - Actions variable-specific to prevent inadvertent flagging of wrong values
  - Right-click/drag to assign, left-click/drag to clear
- Anomaly reports can be auto-generated on demand and annotated to explain rationale for revision



Interactive Q/C Tools

- Composite flags can be manually propagated to derived variables
  - Flags can be meshed with or over-write existing flags
  - Often easier to propagate flags than compose multi-column rule sets

- Whenever flags interactively edited, automatic Q/C rules “locked” to prevent over-riding edits



Software for Q/C Analysis

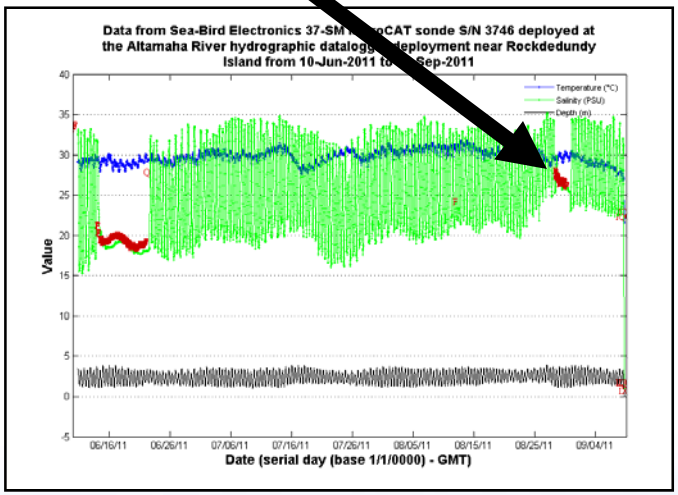
- Programmatic/Algorithmic Q/C Analysis
  - Based on “rules” – expressions evaluated to identify values matching criteria
  - Unlimited Q/C rules can be defined for each attribute
  - Scope can range from single value to entire data set (+ external files)
  - Rules evaluated when data loaded and when data or rules change
  - Rules can be predefined in metadata templates to automate Q/C on import
- Interactive Q/C Analysis and Revision
  - Qualifiers can be assigned/cleared visually on data plots
  - Qualifiers can be propagated to dependent columns en masse
  - Qualifiers can be removed or edited (search/replace) if standards change
- Automatic Documentation of Q/C Steps
  - All Q/C operations (including revisions) logged to processing lineage
  - Data anomalies reports can be auto-generated and annotated to capture rationale

Q/C Framework Components

- Generalized tabular data model designed to support Q/C
  - Any size data table (numeric & text fields)
  - Detailed metadata (dataset-level documentation & attribute descriptors)
  - Q/C rules for every attribute, qualifier flags for every value
  - Data processing and Q/C operation history (lineage)
- Software for Q/C analysis and qualifier flag management
  - Automatic (rule-based) and manual (visual) assignment of Q/C qualifier flags
  - Interactive Q/C qualifier propagation and revision
  - Transparent management of flags throughout all data manipulation
- Software for Q/C-aware data analysis, synthesis
  - Qualified values can be filtered, summarized, visualized during analysis
  - Statistics about missing/qualified values tabulated, used to qualify derived data

Q/C-Aware Data Management & Analysis

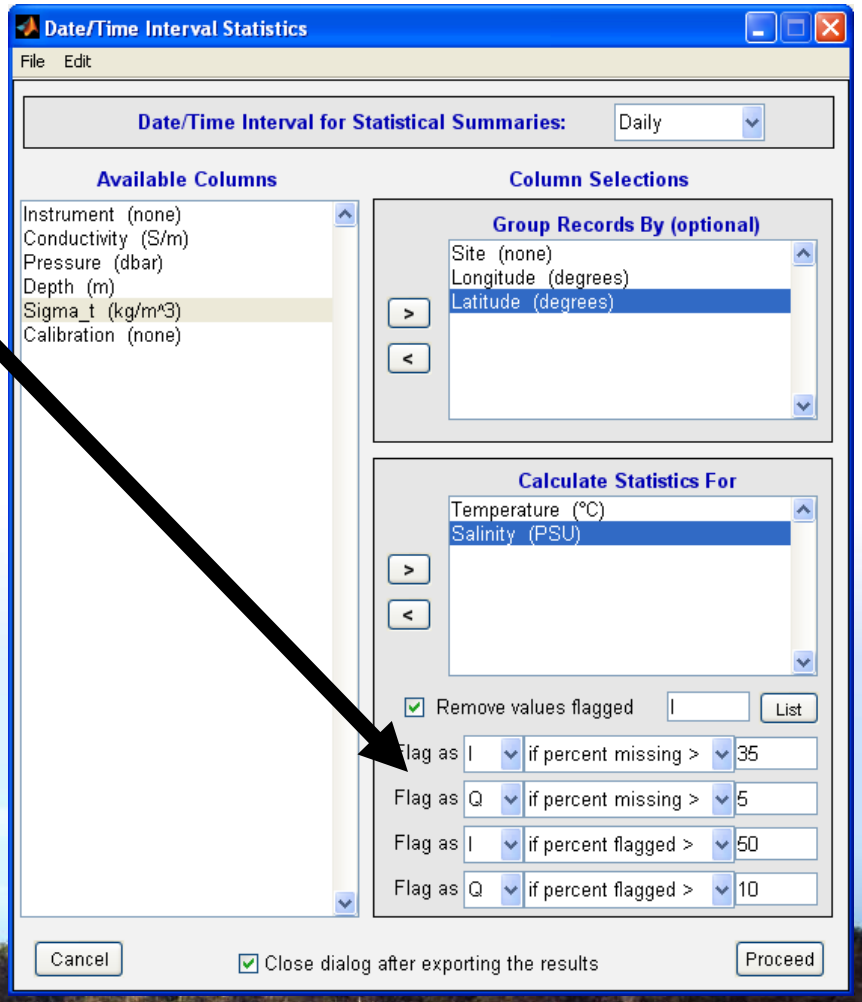
- Q/C flags can be visualized in data editor grid and plots



- Statistical summaries can be generated with/without flagged values
- Flagged, missing values can be summarized by parameter/date
- Flags can be instantiated as coded text columns for export
- Flagged values can be selectively removed from data sets

Q/C-Aware Data Synthesis

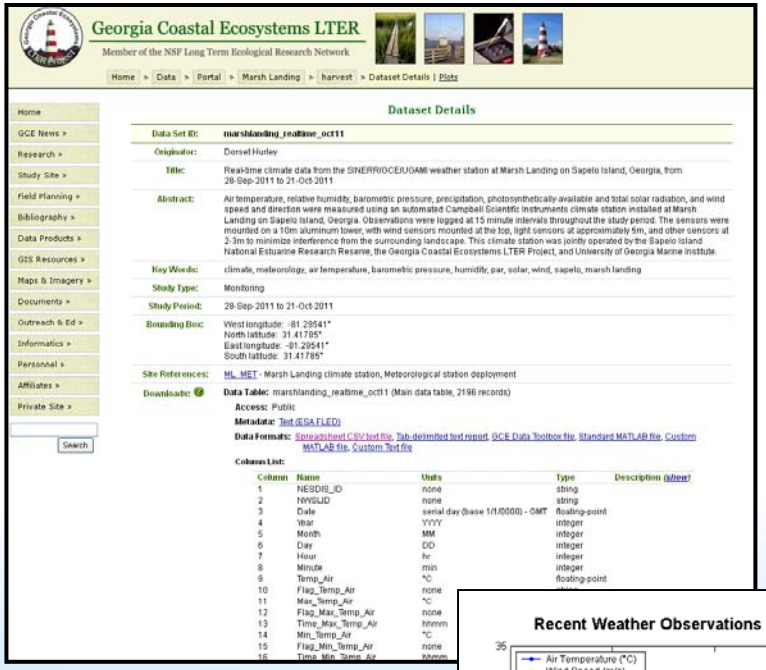
- Flagged, missing values summarized in re-sampled data (aggregated, binned, date-time re-sampled), with automatic Q/C rule creation
- Flags automatically “locked” when merging multiple data
- Flags, rules and definitions move with data when performing relational joins between data sets

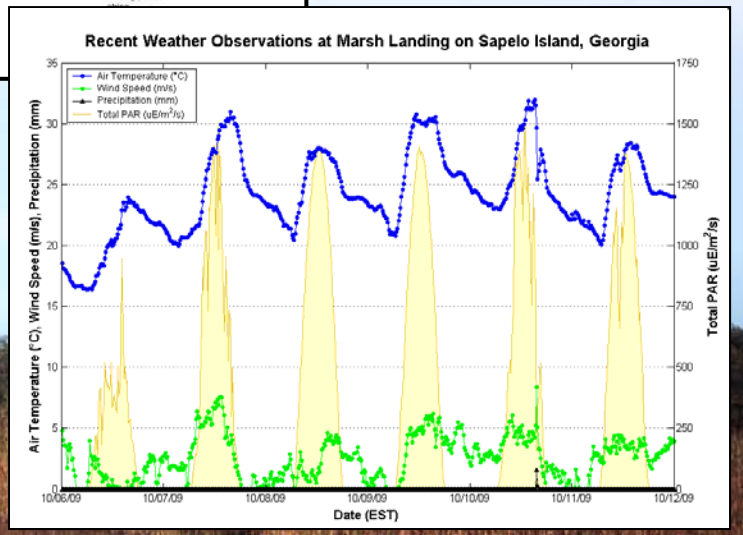


Suitability for Real-Time Sensor Data

- Good Scalability
  - Data volumes only limited by computer memory (tested >2 GB data sets)
  - Multiple instances can be run on high-end, 64bit, clustered workstations
  - Good flag evaluation performance in use, testing with diverse rule sets
- Good scope for automation
  - Command-line API for unattended batch processing via workflow scripts
  - Timed and triggered workflow implementations easy to deploy
- Support for multiple I/O formats, transport protocols
  - Formats: ASCII, MATLAB, SQL, specialized (CSI, SBE, NWIS RDB, HADS, …)
  - Transport: local file system, UNC paths, HTTP, FTP, SOAP
- Already used for real-time GCE data, USGS data harvesting service (LTER HydroDB, CWT)

Real-Time GCE Data Harvesting





Georgia Coastal Ecosystems LTER Project

Implementation Scenarios

- End-to-End Processing (logger-to-scientist)
  - Acquire raw data from logger, file system, network (CIFS,HTTP,FTP,SOAP)
  - Assign metadata from template or using forms to validate and flag data
  - Review data and fine-tune flag assignments
  - Generate distribution files & plots, archive data, index for searching
  - Scientists can use toolbox on their desktop
- Data Pre-processing
  - Acquire, validate and flag raw data (on demand or timed/triggered)
  - Upload processed data files (e.g. csv) or value & flag arrays to RDBMS (e.g HIS)
- Workflow Step
  - Call toolbox from other software as part of workflow (e.g. LoggerNet)
  - Kepler via MATLAB actor
  - DataTurbine via MATLAB off-ramp or Java API

Concluding Remarks

- "Fine Print"
  - Requires MATLAB ($ academic, $$$ government/industry)
  - Software documented, but tutorial and training materials needed (planned)
  - Support is limited (unfunded outreach)
- Benefits
  - Fully cross-platform (Windows, MacOS, Linux, Solaris)
  - Mature – used 24/7 for over 10 years for LTER data management (>3000 dl's)
  - GCE Data Toolbox is free and open source (GPL) – can customize, redistribute
- More information and downloads at:
  https://gce-svn.marsci.uga.edu/trac/GCE_Toolbox

(This work was supported by NSF grant numbers OCE-9982133 and OCE-0620959)