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

Wade M. Sheldon
Department of Marine Sciences, University of Georgia, Athens, Georgia, USA

Abstract

The volume of monitoring data that can be acquired and managed by Long Term Ecological Research sites and environmental observatories 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 instrumented platforms with real-time data telemetry, are raising the bar even higher. Quality control is often a major challenge with real-time data, though, due to poor scalability of traditional software tools, approaches and analysis methods. Software developed at the Georgia Coastal Ecosystems Long Term Ecological Research 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 synthesis. This paper describes the design and operation of the dynamic, rule-based quality control framework provided by this software, and presents quantitative performance data that demonstrate these tools can efficiently perform quality analysis on million-record data sets using commodity computer hardware.

Key Words: quality control, statistical analysis, real-time data, sensor, MATLAB

Introduction

Quality control is a critical component of environmental data management, particularly for data collected by autonomous sensors. Many factors can affect the quality of sensor data, including calibration drift, biological fouling, electrical noise during data transmission, and mechanical interference from other instruments or mounting hardware (Gentili et al, 2004; Magnaterra et al., 2004). These problems lead to data contamination that can profoundly affect data analysis and skew interpretation. Traditionally, quality control of environmental data has been conducted by visually inspecting or plotting data values and performing detailed statistical analyses (e.g. distribution tests, outlier tests) using specialized software (Edwards, 2000). However, the sheer number of parameters and volume of data generated by modern sensors and sensor networks often precludes this approach. Consequently, some monitoring programs (e.g. U.S.G.S. National Water Information System) report provisional near-real-time data with no or minimal quality control processing, then release reviewed, derived data products at a later time.

Software developed at the Georgia Coastal Ecosystems Long Term Ecological Research Site (http://gce-lter.marsci.uga.edu), the GCE Data Toolbox for MATLAB, includes a dynamic, metadata-based quality control framework that has proven useful for analysis of real-time sensor data, as well as non-real-time and legacy data sets. Although other metadata-based quality control processing approaches have been advanced (Nottrott et al., 1999), this software provides a fully integrated, extensible solution that supports both automated and interactive analysis within a seamless software environment. An unlimited number of quality control “rules” can be defined for each parameter in a data set, and rules are evaluated automatically whenever data are

imported or revised to generate alphanumeric "flags" that are intrinsically managed along with the data values they qualify. This paper describes the design and operation of the quality control framework provided by this software, and its potential use for high volume sensor data sets.

Methods and Techniques

The GCE Data Toolbox software package (Sheldon, 2002) was developed using the MATLAB® technical programming language (The MathWorks, http://www.mathworks.com). MATLAB was selected because of its prevalence in environmental science and engineering as well as its excellent support for large data sets (limited only by computer memory) and code portability across Windows, UNIX and Macintosh computer platforms. MATLAB is also a dynamically-typed, interpreted language, making it well suited for rapid software development, testing and deployment (Prechelt, 2000).

The GCE Data Toolbox intrinsically supports data quality control at all levels, starting with the underlying data model (fig. 1). Data sets are managed by toolbox programs as

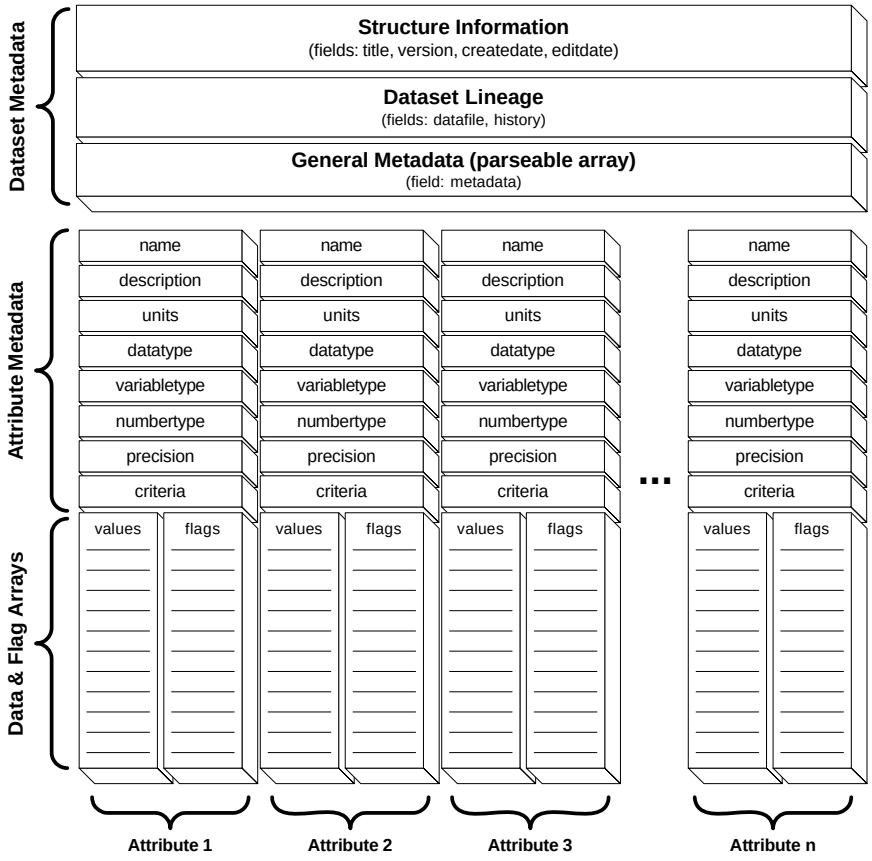

Figure 1. Conceptual model of the GCE Data Structure (version 1.1, 29-Mar-2001), illustrating the types and cardinalities of metadata fields, data arrays and quality control flag arrays. Structures are created and managed using the GCE Data Toolbox, a MATLAB software library for metadata-based analysis, visualization and management of ecological data sets.

MATLAB structure arrays, with dedicated fields for data set metadata and lineage, and repeating groups of attribute metadata fields, data arrays, and quality control flag arrays, which are managed collectively as data set attributes. Correspondence of data values and flags is maintained across attributes throughout all data manipulation operations (e.g. sorting, filtering, joins, unions) similarly to tuples in a relational database model. Flags are stored as single-character alphanumeric codes, which are defined in the data set metadata. An empty string denotes absence of a flag, and multiple flags can be assigned to a single data value.

Quality control rules are defined using the syntax: [expression]='[flag code]', where [expression] is any MATLAB statement that returns a logical array of 1's and 0's, and [flag code] is the alphanumeric character to assign to values matching the criteria (i.e. [expression] evaluates as 1). Data columns are referenced in rules using "x" to represent the current column, or "col_[column name]" to reference any column in the data set by name. For example, the rule "x<0='Q'" assigns "Q" flags to any negative values in the corresponding data column, and "col_Dry_Weight<(col_Wet_Weight-col_Ash_Weight)*0.90 ='I'" (in column Dry_Weight) assigns "I" flags to any values of Dry_Weight that are less than 90% of the difference between Wet_Weight and Ash_Weight. Compound rules can be defined by separating multiple rule expressions with semicolons (e.g. "x<0='I';x>100='I'; x<20='Q';x>80='Q'"). Rule statements are stored in the "criteria" metadata field for each attribute, and can be defined in advance using metadata templates or created and edited interactively using a GUI form (fig. 2).

Rules can be defined to perform a wide variety of quality control analyses based on numeric, text and statistical comparisons using this simple syntax (Table 1). The default framework can also be extended simply by adding custom MATLAB functions to the toolbox directory and referencing these functions in quality control rules. Custom functions can be written to retrieve reference data from a file system, database query or web service then run complex algorithms or models implemented using MATLAB or another programming language (e.g. Java or FORTRAN), so the potential scope of rules is unlimited. It should be noted, though, that although powerful, this feature does represent a potential security risk. Overt system calls in rules are blocked and error handling routines prevent syntax and run-time errors from halting the program or corrupting data, but a malicious user could inject calls to external functions capable of altering data or launching attacks, so access to data set metadata and templates should be controlled in a network setting.

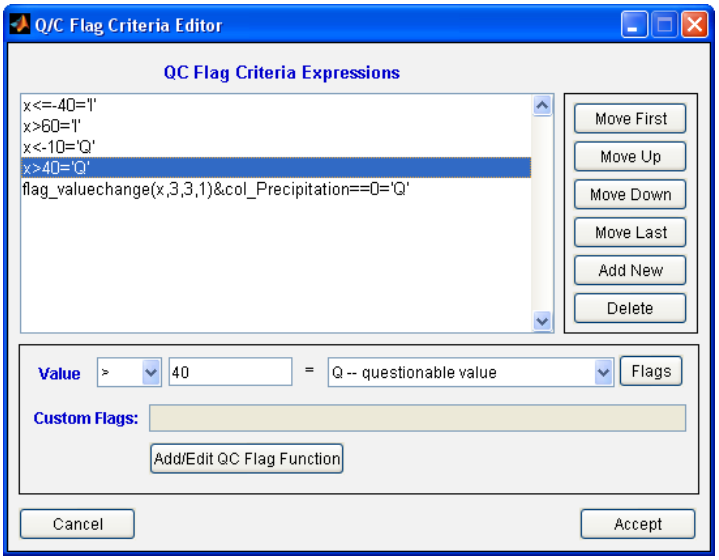

Figure 2. Graphical quality control rule editor form in the GCE Data Toolbox for MATLAB

Table 1. Representative quality control operations, rule types and syntax. Note that "x" symbols in rule criteria are aliases for values in the corresponding data column, and that the "col_" prefix denotes values from any data set column referenced by name (including the column containing the rule).

Operation	Quality Control Goal	Rule Type	Example Syntax
Range check	Confirm values are within range of the sensor/parameter	numeric conditional	x<0='I';x>100='I' -- assigns I flags to negative values and values over 100
Consistency check	Confirm values are consistent with other measured parameters or historic maxima/minima	multi-column conditional	col_DOC_Conc>col_TOC_Conc='I' -- assigns I flags to DOC concentrations that exceed total organic carbon concentration
		statistical expression	x>mean(x)+4*std(x)='Q' -- assigns Q flags to values more than 4 standard deviations above the column mean
Vocabulary check	Confirm values conform to a controlled vocabulary (e.g. standard code list)	custom function	flag_notinlist(x,'A1,A2,A3,A4')='Q' -- assigns Q flags to values not in the specified list (or referenced data set)
Dependency check	Confirm measurements were recorded under suitable conditions, based on other parameter observations	multi-column conditional	col_Depth<0.1='Q' -- in column Salinity; assigns Q flags to values recorded when instrument depth was < 0.1m, indicating water emergence
Pattern check	Confirm values do not exhibit temporal or spatial patterns that indicate sensor failure or data contamination	custom function	flag_percentchange(x,25,25,3)= 'S' -- assigns S flags to values that are >25% above or below the mean of the preceding 3 values
Reference data check	Confirm values agree with prior recorded values or reference values	custom function	flag_locationcoords(x,col_Lon, col_Lat,0.2)='I' -- assigns I flags to location codes that differ by more than 0.2km from the registered coordinates, based on corresponding Latitude and Longitude values

Quality control rules are automatically evaluated to assign or clear flags whenever data values are entered, imported or edited (or the rules themselves are revised) using toolbox functions. Flags can also be assigned manually with the mouse on data plots or using a spreadsheet-like GUI editor to augment or revise rule-based flag assignments. Additionally, flags can be parsed from text attributes in data sets, allowing flags assigned by other data management systems to be imported into the toolbox framework. When flags are defined manually or imported, the token "manual" is added to the corresponding "criteria" attribute metadata field. This token locks flags for the data column so manually-assigned flags are not subsequently overridden by automatic rules. Removing the manual token restores automatic flag evaluation.

Quality control rules and flags are constitutive components of the GCE Data Toolbox data model, so most toolbox functions provide explicit options for handling flagged values during post processing and analysis. For example, flags can be displayed, ignored or flagged-values removed when data are plotted, and statistics reports can be generated with and without flagged values. Data export functions provide various options for formatting flags in delimited text and MATLAB files to support other programs and standards, and data integration tools (e.g. union and join functions) provide options for automatically locking flags to prevent inappropriate application of criteria after multiple data sets are combined. In addition, data aggregation, date/time re-sampling, and binning tools optionally create quality control rules for all derived

data columns based on the number or percentage of flagged and missing values in each respective group, date/time interval or bin to provide quality control for derived data products.

In order to test the performance of the GCE Data Toolbox for quality control analysis of high volume sensor array data sets, a 1,000,000 record by 48 column time series data set was compiled from various sources (i.e. equivalent to one year of observations at 30 sec frequency). Three to six quality control rules were defined for each column, including numeric range checks, statistical consistency checks, and multi-column dependency checks, resulting in flags being assigned to 0-14% (mean 4.3%) of values. The test data table was subset into 12, 24 and 48 column tables of varying length, and the time required to evaluate all rules and manage assigned flags using the "dataflag" function was evaluated for two different versions of MATLAB (release 2007b and release 13a/version 6.51) on a Dell® computer with an Intel® Core Duo™ T2500 processor (2.0 GHz clock speed) and 1 GB RAM.

Results and Discussion

In performance testing, quality control rule evaluation time varied linearly with number of records and number of parameters (Table. 2), with both slopes near unity. These results and additional trials with larger tables on computers with up to 4 GB of system RAM indicate that algorithm execution time is directly proportional to table size for a given rule set, software and hardware configuration. Evaluating rules for the complete 1 million record data set required <42 sec on the test hardware, and a 100,000 record data set required <4 sec, indicating that interactive quality control analysis of typical sensor data sets is very practical.

Table 2. Quality control rule evaluation time in seconds versus data set table size. Timings were evaluated using MATLAB releases 13a (R13a) and 2007b (R2007b).

Data Set Records	12 Parameters (30 rules)		24 Parameters (60 rules)		48 Parameters (120 rules)	
	R13a	R2007b	R13a	R2007b	R13a	R2007b
10,000	0.09	0.08	0.19	0.11	0.39	0.22
50,000	0.45	0.25	0.97	0.51	1.81	1.02
100,000	0.97	0.58	1.97	1.16	3.94	2.36
200,000	1.98	1.17	3.97	2.36	7.95	4.77
400,000	4.03	2.41	8.06	4.86	16.20	9.77
600,000	6.11	3.67	12.25	7.36	24.59	14.88
800,000	8.19	4.91	16.39	9.89	32.84	19.89
1,000,000	10.25	6.14	20.56	12.38	41.08	26.58

The GCE Data Toolbox was developed as a comprehensive data processing solution for GCE information management staff, but it has proven useful beyond this scope. Several GCE investigators and many graduate students have used the toolbox to analyze core GCE data as well as their own data, and the toolbox is used in Marine Science methods classes at UGA. Since its initial release in 2002, over 2800 web visitors not affiliated with the GCE project have downloaded the toolbox for a wide-ranging set of applications (e.g. hydrological data analysis, quality control of U.S.A.F. test flight data, U.S.G.S. and LTER ClimDB data mining). Although formal usability testing has not been performed to date, various enhancements requested by users have been implemented and user feedback on functionality has been uniformly positive.

Conclusions

The GCE Data Toolbox is well suited for processing high volume, real-time sensor array data and performing quality control analysis. Metadata templates containing detailed attribute descriptors and quality control rules can be defined using GUI forms for each sensor platform, and then applied automatically to validate and flag data values when raw data are imported from data loggers, text files, or database queries. Data summaries and plots can be generated to review the quality control analysis results, then flag rules and flag assignments can be refined as necessary. Derived data products (containing additional quality control rules) can then be generated for distribution or further analysis, preserving information about the quality and completeness of the source data in the data set metadata, derived attributes, and quality control rules. This workflow can then be automated for routine data acquisition and analysis.

The performance results reported above indicate that the flag evaluation algorithms are suitably efficient for processing million-record data sets in real time on commodity computer hardware, with maximum data set size only limited by available system RAM. Multiple instances of the GCE Data Toolbox can also be run on a single computer to simultaneously process multiple data streams, minimizing the MATLAB licenses required to use the software.

A compiled version of the toolbox is publicly available online (Sheldon, 2002), and source code is available on request for evaluation and end-user customization. Offers to collaborate on future toolbox development are also welcome.

Acknowledgments

This material is based upon work supported by the National Science Foundation under grant numbers OCE-9982133 and OCE-0620959.

References

Edwards, D. 2000. Data Quality Assurance. Pg. 70-91 in: Michener, W.K. and Brunt, J.W., eds. Ecological Data - Design, Management and Processing. Blackwell Scientific, Oxford.

Gentili, S., Magnaterra, L. and Passerini, G. 2004. An introduction to the statistical filling of environmental data time series. Pg. 1-27 in: Latini, G. and Passerini, G., eds. Handling Missing Data: Applications to Environmental Analysis. WIT Press, Boston.

Magnaterra, L., Passerini, G. and Tascini, S. 2004. Data validation and data gaps in environmental time series. Pg. 29-89 in: Latini, G. and Passerini, G., eds. Handling Missing Data: Applications to Environmental Analysis. WIT Press, Boston.

Nottrott, R., Jones, M.B. and Schildhauer, M.P.1999. Using XML-Structured Metadata to automate quality assurance processing for ecological data. Proceedings of the Third IEEE Computer Society Metadata Conference. IEEE. Bethesda, MD.

Prechelt, L. 2000. An Empirical Comparison of Seven Programming Languages. IEEE Computer, 33(10):23-29.

Sheldon, W.M. 2002. GCE Data Toolbox for MATLAB – Software tools for metadata-based analysis, visualization and transformation of ecological data sets. (http://gce-lter.marsci.uga.edu/public/im/tools/data_toolbox.htm)