

A Query Language for Retrieving Information from Raw Data Files

Michael W. Senko and Eric C. Hemenway

Thermo Fisher Scientific, San Jose, California

Thermo
SCIENTIFIC

Overview

Purpose: To simplify access to the contents of raw data files

Methods: Layering a simple query language on top of normal data file access libraries

Results: Flexible access to raw file contents using simple query statements

Introduction

Although there are numerous applications for evaluating mass spectral data, users often have novel data evaluation requirements which are not met by readily available programs. Custom applications can always be created, with the challenge often not in the processing algorithm, but in retrieval of information from proprietary format files using the vendor supported libraries. Here we describe the implementation of a simple query language, Xcalibur™ Query Language (XQL), and associated applications which significantly reduce the barrier to accessing raw data. This results in the raw file appearing as if it were contained in a relational database, without the necessity of converting the file to a database.

Methods

XQLConsole is a command line application written in Borland® C++ Builder® 6.0. This program accesses the contents of the raw data file with standard function calls using the XRawFile2 dynamic link library which is part of the Xcalibur Development Kit (XDK). The entire application consists of ~2,000 lines of code.

A restricted subset of the Structured Query Language (SQL) syntax is implemented with a small parser and interpreter using the familiar SELECT ... FROM ... WHERE notation. The interpreter converts the XQL queries from the user and extracts the necessary information from the data file using the standard data access procedures. The results of the query are output in text format using tab delimited columns to either standard out or a named file.

The "SELECT" operator is used to specify the fields which should be returned from the raw data file. These include fields for the file, scan header, scan filter, scan trailer, spectrum, and status information. The wildcard "*" can be used to retrieve all information (excluding scan data) from specified scans. All currently supported fields are listed in Table 1.

The fields for the trailer and status information are specific to the instrument and version of software which produced the raw data file. Therefore, these fields are user-configurable using a simple text file to support future expansion.

The "FROM" operator specifies the raw file(s) from which to retrieve information. This can either be a comma separated list of raw files, or may be "*.raw", which allows selection of all raw files in the current working directory.

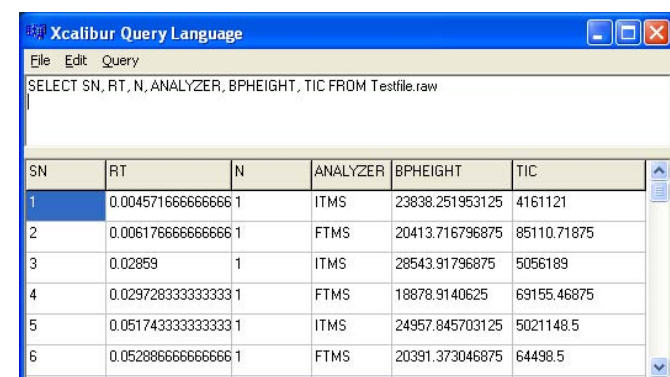
TABLE 1. Available fields for use with the SELECT operator

File Fields	Filter Fields	Default Trailer Fields	Spectrum Fields
FILENAME	FILTER	EVENT	MASS
PATHNAME	ANALYZER	SEGMENT	INTENSITY
TABLEID	POLARITY	ST	MASSBP
SCANCOUNT	DATATYPE	Z	INTENSITYBP
Header Fields	N	MONO	LMASS
SN	PRECURSOR	MASTER	LINTENSITY
TIC	SOURCE	IT	LMASSBP
RT			LINTENSITYBP
BPHEIGHT			AREA
BPMASS			
FM			
LM			
PAIRS			

The "WHERE" operator specifies conditions for filtering the retrieved information. The operators that are supported are =, !=, <=, >=, < and >. All fields previously described may be used as operands along with any literal values. The functions may be logically linked with "AND" and "OR" statements and may be grouped with parentheses to set operator precedence.

The results generated by the console application can be imported by any program that is able to read tab separated text files. A separate GUI, XQLProject, has been created in Borland C++ Builder to simplify interfacing with the command line application (Figure 1). This provides a text box for inputting the XQL-formatted query and provides a limited grid for displaying the results. The grid is tied to the standard clipboard, and provides a route to export the results to any Windows® program such as Microsoft® Excel® for further analysis and plotting.

FIGURE 1. Screen Shot of the XQLProject application, which provides a graphical user interface for the command line application.



The screenshot shows the Xcalibur Query Language application. The query window contains the text: `SELECT SN, RT, N, ANALYZER, BPHEIGHT, TIC FROM Testfile.raw`. Below the query window is a table with the following data:

SN	RT	N	ANALYZER	BPHEIGHT	TIC
1	0.0045716666666666	1	ITMS	23838.251953125	4161121
2	0.0061766666666666	1	FTMS	20413.716796875	85110.71875
3	0.02859	1	ITMS	28543.91796875	5056189
4	0.0297283333333333	1	FTMS	18878.9140625	69155.46875
5	0.0517433333333333	1	ITMS	24957.845703125	5021148.5
6	0.0528866666666666	1	FTMS	20391.373046875	64498.5

Results

XQL treats raw data files in a directory as tables of a database and provides data access through the construction of queries. XQL acts as a simple high-level abstraction layer on top of the complex low-level file access libraries supplied in vendor software development kits. This reduces the process of retrieving the data points in a mass spectrum to the execution of the query "SELECT MASS, INTENSITY FROM MSData.raw WHERE SN = 100" (Figure 2). An extracted ion chromatogram can be retrieved by execution of the query "SELECT RT, AREA FROM MSData.raw WHERE MASS > 500 AND MASS < 501" (Figure 3).

FIGURE 2. An Excel plot of the query "SELECT MASS, INTENSITY FROM MSData.raw WHERE SN = 100" plotted in Excel

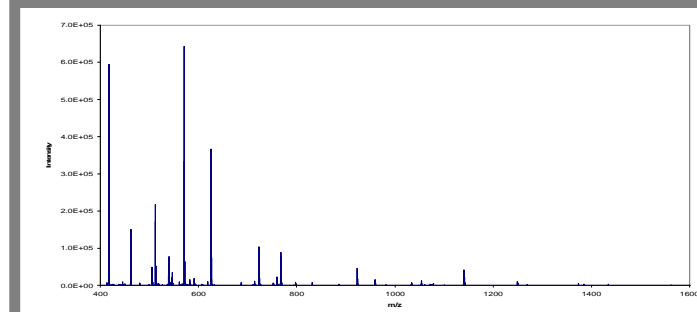
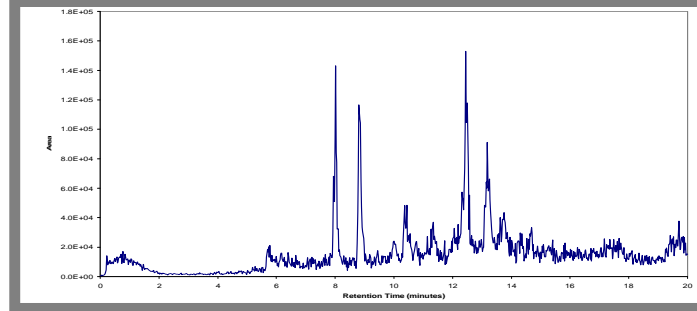
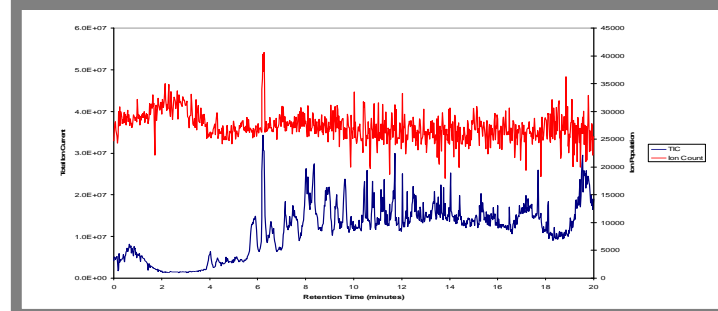


FIGURE 3. An Excel plot of the query "SELECT AREA, RT FROM MSData.raw WHERE MASS > 500 AND MASS < 501"



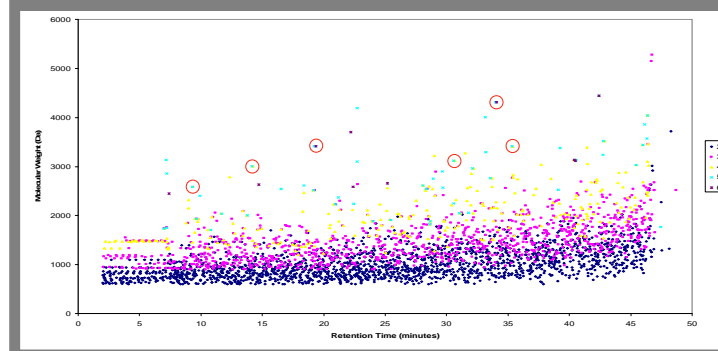
A slightly more practical example is in the evaluation of the performance of Automatic Gain Control (AGC), which is meant to maintain a consistent ion population. The total ion population for an LTQ™ can be calculated by multiplying the Total Ion Current by the injection time in seconds. As shown in Figure 4, the ion population remains fairly constant except in the case where one strong peak eluted and caused saturation of the detector during the prescan.

FIGURE 4. The ion population in the trap can be determined with the query "SELECT RT, TIC, IT FROM MSData.raw"



The final example demonstrates the ability to extract information from the scan trailer. To create a plot of all precursors selected data dependently for MS/MS, separated by the observed charge state, the query would be "SELECT RT, PRECURSOR, Z FROM MSData.raw WHERE N = 2". This plot can show trends in molecular weight, and graphically displays instances where the same peptide was selected more than once based on different charge states.

FIGURE 5. "SELECT RT, PRECURSOR, Z FROM MSData.raw WHERE N = 2"



Conclusions

The use of a query language provides a simple, yet flexible and powerful method for extracting information from raw data files. This eliminates the complexity of learning vendor-specific file access libraries without restricting access to the wealth of information contained stored in these files.

Borland and C++ Builder are registered trademarks of Borland Software Corporation. Microsoft, Excel, and Windows are registered trademarks of Microsoft Corporation. All other trademarks are the property of Thermo Fisher Scientific Inc. and its subsidiaries.