msXpertSuite

the expert massist's
software suite

Simulating and analyzing
ionized flying species

Suite version 4.1.0

By Filippo Rusconi, Ph.D.
Researcher at the CNRS, France

---

msXpertSuite version 4.1.0

mineXpert

User Manual

This User Manual is distributed at
http://www.msxpertsuite.org

Filippo Rusconi, Ph.D.

Scientist at CNRS

Centre national de
la Recherche scientifique

Université Paris-Sud
Ferme du Moulon
Gif-sur-Yvette
France

***mineXpert* User Manual**

http://www.msxpertsuite.org

Revision History

* **december 2017** Major rewriting of the document to incorporate all the new features. Large chapter on the scripting of *mineXpert*.
* **may 2017** Refactored document. First version documenting almost all the of current features of the software program.
* **november 2016** Resume writing, with new program name: *mineXpert*.
* **september 2016** Start of writing.

Contents

List of Figures

1

Preface

This manual is about the *msXpertSuite* mass spectrometric software suite, a software environment that aims at:

* Letting users predict/analyze mass spectrometric data on (bio)polymers (*massXpert* module);

* Letting user load, view and analyse and mine mass spectrometric data (*mineXpert* module).

As such, this manual is intended for people willing to learn how to install and use the comprehensive *msXpertSuite* software package.

Mass spectrometry has gained popularity across the past twenty years or so. Indeed, developments in polymer mass spectrometry have made this technique appropriate to accurately measure masses of polymers as heavy as many hundreds of kDa, and of any chemical type.

There are a number of utilities—sold by mass spectrometer constructors with their machines, usually as a marketing "plus"—that allow predicting/analyzing mass spectrometric data obtained on polymers. These programs are usually different from a constructor to another. Also, there are as many mass spectrometric data prediction/analysis computer programs as there are different polymer types. You will get a program for oligonucleotides, another one for proteins, maybe there is one program for saccharides, and so on. Thus, the biochemist/massist, for example, who happens to work on different biopolymer types will have to learn to use several different software packages. Also, if the software user does not own a mass spectrometer, chances are he will need to buy all these software packages.

The *msXpertSuite* mass spectrometric software is designed to provide *free* solutions to all these problems by providing the following features:

* *massXpert*:
  - Model *ex nihilo* polymer chemistry definitions (in the *XpertDef* module that is part of the *massXpert* program);
  - Peform simple yet powerful mass computations to be made in a mass desktop calculator that is both polymer chemistry definition-aware and fully programmable (that's the *XpertCalc* module also part of the *massXpert* program);
  - Edit polymer sequences on a polymer chemistry definition-specific basis, along with chemical reaction simulations, finely configured mass spectrometric computations... (all taking place in the *XpertEdit* module that is the main module of the *massXpert* program);
  - Customize the way each monomer will show up graphically during the program operation (in the *XpertEdit* module);
  - Edit polymer sequences with immediate visualization of the mass changes elicited by the editing activity (in the *XpertEdit* module);
  - Open an unlimited number of polymer sequences at any given time and of any given polymer chemistry definition type (in the *XpertEdit* module).

* *mineXpert*:
  - Load mass spectometry data files of the main known open formats (*mzML*, *mzXML*, *txt*, *xy*, thanks to the excellent `libpwiz` library of ProteoWizard[1] fame);
  - Display the data in powerful ways in a unified graphical user interface. The interface was designed to integrate all the most useful characteristics of the various proprietary environments known by the author (thanks to the excellent `libqcustomplot` library[2];
  - Perform data mining by performing data integrations in various ways;
  - Ion mobility mass spectrometry data are handled;
  - Data integration allows easy quantitation of spectral data at any level (TIC chromatogram, mass spectrum, drift spectrum);
  - Innovative data analysis recording allows to store the features mined during the data mining sessions in flexible ways that allow further data processing, like injection in databases;
  - Convert data from mzML to the private (albeit open) database file format that allows to load data much faster. *mineXpert* can slice big data files into smaller chunks retaining all the data selected by the user in the most flexible ways.

[1] http://proteowizard.sourceforge.net/
[2] http://qcustomplot.com/.

Project History

This is a brief history of *msXpertSuite*.

* **1998–2000** The name *massXpert* comes from a project I started while I was a post-doctoral fellow at the École Polytechnique (Institut Européen de Chimie et Biologie, Université Bordeaux 1, Pessac, France).

  The *massXpert* program was published in *Bioinformatics* (Rusconi, F. and Belghazi, M. *Desktop prediction/analysis of mass spectrometric data in proteomic projects by using massXpert ˙ Bioinformatics*, 2002, 644–655).

  At that time, *MS-Windows* was at the *Windows NT 4.0* version and the next big release was going to be "you'll see what you'll see" : *MS-Windows 2000*.

  When I tried *massXpert* on that new version (one colleague had it with a new machine), I discovered that my software would not run normally (the editor was broken). The Microsoft technical staff' would advise to "buy a new version of the compiler environment and rebuild". This was a no-go: I did not want to continue paying for using something I had produced.

* **2001–2006**

  During fall 1999, I decided that I would stop using Microsoft products for my development. At the beginning of 2000 I started as a CNRS research staff in a new laboratory and decided to start fresh: I switched to GNU/Linux (I never looked back). After some months of learning, I felt mature to start a new development project that would eventually become an official GNU package: GNU polyxmass.

  The GNU polyxmass software, much more powerful than what the initial *massXpert* software used to be, was published in *BMC Bioinformatics* in 2006 (Rusconi, F., *GNU polyxmass: a software framework for mass spectrometric simulations of linear (bio-)polymeric analytes. BMC Bioinformatics*, 2006,226).

  Following that publication I got a lot of feedback (very positive, in a way) along the lines: —*"Hey, your software looks very interesting; only it's a pity we cannot use it because it runs on GNU/Linux, and we only use MS-Windows and MacOSX!"*

* **2007–2016**

  In december 2006, I decided to make a full rewrite of GNU polyxmass. The software of which you are reading the user manual is the result of that rewrite. I decided to "recycle" the *massXpert* name because this software is written in C++, as was the first *massXpert* software. Also, because the first *MS-Windows*-based *massXpert* project is not developped anymore, taking that name was kind of a "revival" which I enjoyed. However, the toolkit I used this time is not the Microsoft Foundation Classes (first *massXpert* version) but the Trolltech Qt framework (see the "About Qt" help menu in *massXpert*).

  Coding with Qt libraries has one big advantage: it allows the developer to code once and to compile on the three main platforms available today: *GNU/Linux*, *MacOSX*, *MS-Windows*. Another advantage is that

Qt libraries are wonderful software, technically and philosophically (Free Software).

* **2016–**

  In 2016, I started a new project about visualization of mass spectrometric data. The project developed pretty quickly, as we needed at the mass spectrometry facility a software that would allow to cope efficiently with ion mobility mass spectrometric experimental data. *mineXpert* was thus started.

  To bundle both *massXpert* and *mineXpert* in a single software suite, I bought the *msXpertSuite* website `http://msxpertsuite.org` and created that new name.

Typographical Conventions

Throughout the book the following typographical conventions are used:

* *emphasized text* is used each time a new term or concept is introduced
* `shell-prompt $` shows the prompt at which a command should be entered as non-root
* `shell-prompt #` shows the prompt at which a command should be entered as root
* `this typography` applies to commands that the user enters at the shell prompt along with eventual options
* ↩ symbolizes pressing the Enter key
* `this typography` applies to an output resulting from entering a command at the shell prompt
* `emacs` or `libQtCore` names of a program or of a library
* `KDE`, `The Gimp` is the name of a generic software (not a specific executable file)
* `/usr/local/share/massxpert`, `/usr/bin/massxpert` are names of a directory or of a file
* `http://www.gnu.org` is an URL (Uniform Resource Locator)

Program Availability, Technicalities

The ancestor of *massXpert*, GNU polyxmass, was initially developed on a *GNU/Linux* system (RedHat distribution versions successively 6.0, 7.0, 7.2, 7.3, 8.0, 9.0) using software from the Free Software Foundation (FSF[3]). The main

[3] For an in-depth coverage of the philosophy behind the FSF, specifically creating a *free operating system*, you might desire to visit `http://www.gnu.org`.

libraries used were `libglib`, `libgobject`, `libxml2` and `libgtk+`. Since mid-2002, the development was performed on a *Debian GNU/Linux* system (http://www.debian.org), which I find to be the ultimate highly-configurable easy-to-use distribution. *massXpert* is still developed using the *Debian GNU/Linux* system, using Free Software libraries that allow cross-platform computer program development with unprecedented ease (Qt libraries[4]). Developing for *GNU/Linux* has been utterly exciting and extremely efficient.

msXpertSuite' Licensing

The front matter of this manual contains a Copyright statement. I retain the copyright to *msXpertSuite* and all related writings (source and configuration files, programmer's documentation, user manual...) I encourage others to make copies of the work, to distribute it freely, to modify the work and redistribute that derivative work according to the GNU General Public License version 3. The aim of this licensing is to favor spread of knowledge to the widest public possible. Also, it encourages interested hackers[5] to change the code, to improve it and to send patches to the author so that their improvements get into the program to the benefit of the widest public possible. For an in-depth study of the FREE SOFTWARE philosphy I kindly urge the reader to visit http://www.gnu.org/philosophy.

[4] Originally from the Trolltech company (http://www.trolltech.com); now The Qt Company Ltd (doc.qt.io).

[5] *Hacker* is a specialized term to design the programmer who codes programs; this term should *not* be mistaken with *cracker* who is a person who uses computer science knowledge to break information systems' security barriers.

Contacting The Author

msXpertSuite is the fruit of years of work on my part.[6] While I've put a lot of energy into making this program as stable and reliable a piece of software as possible, *msXpertSuite* comes with no warranty of any kind.

The public home page is at http://www.msxpertsuite.org.

The code repository is at https://sourcesup.renater.fr/projects/msxpertsuite.

The Git repos page is at https://sourcesup.renater.fr/scm/browser.php?group_id=3534

Interaction with the developer(s) and the user(s) is according to two modes:

* **Forum** https://sourcesup.renater.fr/forum/forum.php?forum_id=4987
* **Mailing lists**
  - `msxpertsuite-commits@groupes.renater.fr` (subscribe at msxpertsuite-commits-request@groupes.renater.fr);
  - `msxpertsuite-discussion@groupes.renater.fr` (subscribe at msxpertsuite-discussion-request@groupes.renater.fr);

To direct any comment(s) to the author through snail mail, use the following address:

Dr Filippo Rusconi

Génétique Quantitative et Évolution & Plateforme PAPPSO
UMR CNRS 8120 - INRA - Université Paris-Sud - AgroParisTech - Université Paris-Saclay
http://moulon.inra.fr/ & http://pappso.inra.fr/
Ferme du Moulon
91190 Gif-sur-Yvette
France
Fax : +33 (0)1 69 33 23 40

[6] As said earlier, *msXpertSuite* is the successor to the *massXpert* project of which it inherits all the original features, while still integrating new interesting developments.

2

mineXpert Generalities

In this chapter, I wish to introduce some general concepts around the *mineXpert* program and the way data elements are called in this manual and in the program.

A mass spectrometry experiment generally involves monitoring the m/z value of analytes injected in the mass spectrometer along a certain time duration. The m/z value of each detected analyte is recorded along with the corresponding signal intensity (i), so that a mass spectrum is nothing but a series of (mz , i) pairs recorded along the acquisition duration. All along the acquisition, the precise moment at which a given analyte is detected (and its (mz , i) pair is recorded), is called the retention time of that analyte. This retention time is not to be misunderstood as the drift time of that analyte in an ion mobility mass spectrometry experiment.

General Concepts and Terminologies

Most generally, the mass spectrometer acquires an important number of spectra in, say, one second. But all these spectra are *combined* together, and, on the surface, the massist only sees a "slow" acquisition of 1 spectrum per second. This apparent slow acquisition rate is configurable. At the time of writing, generally 1 spectrum per second is recorded on disk. So, say we record mass spectra for 5 minutes, we would have recorded (5*60) spectra.

Acquiring Mass Data Along Time: To Profile or Not To Profile?

As a mass spectrometry user, the reader of this manual certainly has used mass spectrometers where mass spectra are acquired and stored in different ways:

* Mass spectra are acquired and summed—the next to the previous—in such a manner that one is left, at the end of the acquisition, with a single spectrum of which the various peak intensities have been increasing all along the acquisition. Indeed, in this mode, each new spectrum is actually "*combined*" to the previously acquired ones. The resulting mass spectrum that is displayed on screen and that gets ultimately stored on disk is called a *combined spectrum*. This is typically the way MALDI-TOF mass spectrometers are used when acquiring data from samples deposited onto sample plates. We refer to this kind of acquisition as an *"accumulation"* mode acquisition;

* Mass spectra are acquired and stored on disk as a single file containing all the spectra, appended one after the other. There is no combination of the spectra: each time a new spectrum is displayed on screen, that spectrum is appended to the file[1]. This is typically the case when mass spectra are acquired all along a chromatography run and is generally called a *"profile"* mode acquisition.

Mass Data Visualisation: To Combine or Not To Combine?

In the previous section, we mentioned *spectrum combination* a number of times. What does that mean, that spectra are *"combined"* together into a single *"combined spectrum"*? Say we have 200 spectra that need to be combined together into a *single* spectrum that summatively represents the data of these 200 spectra.

First, a new spectrum would be allocated *(result spectrum)*, entirely empty at first. Then, the very first spectrum of the 200 spectra is literally copied into that result spectrum. At this point the combination occurs, according to an iterative process that has the following steps:

* Pick the next spectrum of the 200 spectra dataset;

    1. Pick the first (mz , i) pair of the currently iterated spectrum;
    2. Look up in the *result spectrum* if a m/z value identical to the m/z value of the current (mz , i) pair is already present;
    3. If the m/z value is found, increment its intensity by the intensity of the (mz , i) pair;

[1] Although there certainly *is* spectrum combination going on in the guts of the software, because the system actually acquires much more spectra than is visible on screen and each newly displayed spectrum is actually the combination of many spectra acquired under the surface.

4. Else, if the m/z value is not found, add the current (mz , i) pair to the result spectrum;
5. Iterate over all the remaining (mz , i) pairs of the current spectrum and redo these steps.

* Iterate over all the 198 remaining spectra of the dataset and do the steps above for each single iterated spectrum.

At the end of the two nested loops above, the combined spectrum is still a single spectrum that represents—summatively—all the 200 spectra. This whole process is very computing-intensive, in particular if:

* The m/z range is large: there are lots of points in each spectrum, which means that for each new (m/z , i) pair we need to iterate in the long list of m/z values that make the result spectrum;
* The resolving power of the mass spectrometer is high: there are many points per m/z range unit.

When a profile mode acquisition is performed, the user gets an innumerable number of distinct spectra, all appended to a single file. These unitary spectra are virtually unusable if an initial processing is not performed. This initial processing of the spectra is called *"total ion current chromatogram calculation"*. What is it? Let's say that the user has performed a profile mode mass spectrometry acquisition on the eluate of a chromatography column. Now, imagine that the spectrometer stores the mass data at a rate of one spectrum per second and that the chromatography gradient develops over 45 min: there would be a total of (45 * 60) spectra in that file. The question is: –*"How can we provide the user with a data representation that might be both meaningful and useful to start mining the data?"* The conventional way of doing so is to load all the mass spectra and compute the *"total ion current chromatogram"* (the TIC chromatogram). The analogy with chromatography is evident: the TIC chromatogram is the same as the UV chromatogram unless optical density is not the physical property that is measured over time; instead, the amount of ions that are detected in the mass spectrometer is measured over time. That amount is actually the sum of the intensities of all the (mz , i) pairs detected in each spectrum. When mass data are acquired during a chromatography run, often, the total ion current chromatogram mirrors (mimicks) the UV chromatogram[2]. For each retention time (RT) a TIC value is computed by summing the intensities of all the (mz , i) pairs detected at that specific RT.

How is this total ion current chromatogram computed? This is an iterative process: from the first spectrum (retention time 0 s), to the second spectrum (retention time 1 s) up to the last spectrum (retention time 45 min), the program computes the sum of the intensities of all the spectrum's (mz , i) pairs. That computation ends up with a map that relates each RT value with the corresponding TIC value. The TIC chromatogram is nothing but a plot of the TIC values as a function of RT values. In that sense, it is indeed a chromatogram.

mineXpert works exactly in this way. When mass spectrometry data are loaded from a file, the TIC chromatogram is computed and displayed. This TIC

[2]Unless eluted analytes *do* absorb UV light but *do not* either desorb/desolvate or ionize, or both.

chromatogram serves as the basis for the mass data mining, as described in this manual. The TIC chromatogram serves as the basis for spectral combinations that can be performed in various ways, and not all formally *combinations*, which is why I prefer the term *"integrations"*. Some of these integrations are described below:

* Integrating data from the TIC chromatogram to a single mass spectrum;
* Integrating data from the TIC chromatogram to a single drift spectrum;

Note that the reverse actions are possible (and indeed necessary for a thorough data mining): selecting a region of a mass spectrum and asking that the TIC chromatogram be reconstituted from there; or selecting a region of a drift spectrum and asking that the TIC chromatogram be reconstituted from there also. Finally, integrations may, of course, be performed from a mass spectrum to a drift spectrum, and reverse.

Examples of Various Integrations

In the sections below, the inner workings of *mineXpert* are described for some exemplary mass data integrations. For example, when doing ion mobility mass spectrometry data mining, it is essential to be able to characterize most finely the drift time of each and any analyte. Since each analyte is actually defined as one or more (mz , i) pairs, it is essential to be able to ask questions like the following:

* What is the drift time of the ions below this mass peak[3]?
* What are all the drift times of all the analytes going through the mobility cell for a given retention time range?
* What are all the ions that are responsible for this shoulder in the drift spectrum?

TIC→MZ integration

What computation does actually *mineXpert* do when a mass spectrum is computed starting from a TIC chromatogram region, say between retention time RT minute 7 and RT minute 8.5?

1. List all the mass spectra that were acquired between RT 7 and RT 8.5. In this *spectral set*, there might be many hundreds of spectra that match this criterion, if we think that in ion mobility mass spectrometry ≈ 200 spectra are acquired and stored individually every second (I mean it, every 1 s time lapse);
2. Allocate a new empty spectrum—the "*combined spectrum*"—and copy into it without modification the first spectrum of the spectral set;

[3] Be that mass peak an isotopic cluster or the average envelope of a large analyte at a given charge.

3. Go to the next spectrum of the spectral set and iterate into each (mz , i) pair:

   * Check if the m/z value of the iterated pair is already present in the combined spectrum. If so, increment the combined spectrum's (mz , i) pair's intensity value by the intensity of the iterated (mz , i) pair. If not, simply copy the iterated (mz , i) pair in the combined spectrum;
   * Iterate over all the remaining (mz , i) pairs and perform the same action.

4. Iterate over all the remaining spectra of the spectral set and perform step number 3.

mineXpert then displays the combined spectrum.

TIC→DT integration

What computation does *mineXpert* actually do when a drift spectrum is computed starting from a given TIC chromatogram region, say between retention time RT minute 7 and RT minute 8.5?

What is a drift spectrum? A drift spectrum (mobilogram) is a plot where the cumulated ion current of the detected ions is plotted against the drift time at which they were detected. Let's see how that computation is handled in *mineXpert*.

1. Create a map to store all the (drift time, intensity) pairs that are to be computed below, the (dt,i) map;

2. List all the mass spectra that were acquired between RT 7 and RT 8.5. The obtained list of mass spectra is called the *"spectral set"*;

3. Go to the first spectrum of the spectral set and compute its TIC value (sum of all the intensities of all the (mz , i) pairs of that spectrum). Get the drift time value at which this mass spectrum was acquired. We thus have a value pair: (dt, i), that is, for drift time *dt*, the intensity of the total ion current is *i*;

   At this point, we need to do a short digression: we saw earlier that, at the time of this writing, one of the commercial instruments on which the author of these lines does his experiments stores 200 spectra each second. These 200 spectra actually correspond to the way the drift cycle is divided into 200 bin (time bins). That means that in the retention time range [7–8.5], there are (1.5*60) complete drift cycles. And thus there are (1.5*60) spectra with drift time x, the same amount of spectra with drift time y, and so on for the reminaing 198 time bins. Of course, a large number of these spectra might be almost empty, but these spectra are there and we need to cope with them.

   The paragraph above must thus lead to one interrogation about the current (dt,i) pair: –*"Has the current dt value be seen before, during the previous iterations in this loop?"* If no, create the (dt, i) pair and add it to the (dt,i) map; if yes, get the *dt* element in the map and increment its intensity value by the TIC value computed above;

4. Iterate over all the remaining spectra of the spectral set and perform step number 3.

At the end of the loop above, we get a map in which each item relates a given drift time with a TIC value. This can be understood this way: –*"For each drift time value, what is the accumulated ion current of all the ions having that specific drift time?"*.

At this point, *mineXpert* displays the drift spectrum (mobilogram).

3

mineXpert: A Powerful Mass Spectrum Viewer

Data mining, in mass spectrometry, entails, for a large part, the relentless scrutinization of the mass spectra by an expert eye. Without a powerful mass spectrum viewer, capable of numerous data display modes, the expert eye remains powerless.

After having completed this chapter you will be able to perform mass spectrum visualization and analysis, optionally reporting all the analysed peaks to a file on disk.

Opening mass spectrum files

To start a *mineXpert* session, open one or more mass spectra using the menu *File⟶Open full mass spectrum file(s)*. The following file formats are understood by *mineXpert*, partly thanks to the `libpwiz` library from the `ProteoWizard` project[1]:

* *mzML,mzXml,MS1,MS2,MGF* files (`libpwiz`);
* *txt,asc* files where m/z and i values are separated by any character that is neither a newline nor a dot nor a digit (loading is handled by a private parser);
* `xy` files from Bruker (private parser);
* *SQLite3*, a private open/documented database format (private parser).

There are two variants of the mass spectrometry file opening menu, one for which *all* the mass data are read from file and stored in memory and one for which the mass data are read from file in streamed mode, used to compute the TIC chromatogram and discarded. The latter mode is useful when the mass data are so large that they cannot fit in memory.

The Window Layout

The graphical interface of *mineXpert* comprises a number of windows where data and informations are displayed. These windows are described below (see Figure 3.1 on page 17):

* mineXpert main program window: this is an unintrusive window sporting the main menu and a status bar where messages are displayed;
* The Loaded mass spectrum files window, that lists all the mass spectrometry data files that are currently loaded in the program;
* The TIC chromatogram window[2] where the various TIC chromatograms are displayed for the various mass spectrometry data files that have been loaded. There is, by definition, a single TIC chromatogram per data file currently loaded in the program. However, this window will also display TIC chromatograms that are computed as an integration step from the other windows, like from the Mass spectrum window or from the Drift spectrum window. In this case, the chromatogram is an extracted ion current chromatogram (XIC chromatogram);
* The Mass spectrum window, where the various mass spectra are displayed. A given mass spectrum may originate from a TIC chromatogram or from a drift spectrum, or even from a color map. A given originating chromatogram or drift spectrum or color map may be the origin of more than one derived mass spectrum;

[1] Please, see `http://proteowizard.sourceforge.net/`
[2] TIC stands for “total ion current”.

* The Drift spectrum window, where the various drift spectra are displayed. Drift spectra can originate from the TIC chromatograms, from the mass spectra or from the color map;
* The Color map window, that contains a single color map for each loaded mass data file. At the time of this writing, there is no way to produce a color map from any other window;
* The Console window, where the various messages or analysis data elements are displayed for the user to select, copy and paste in an electronic labbook;

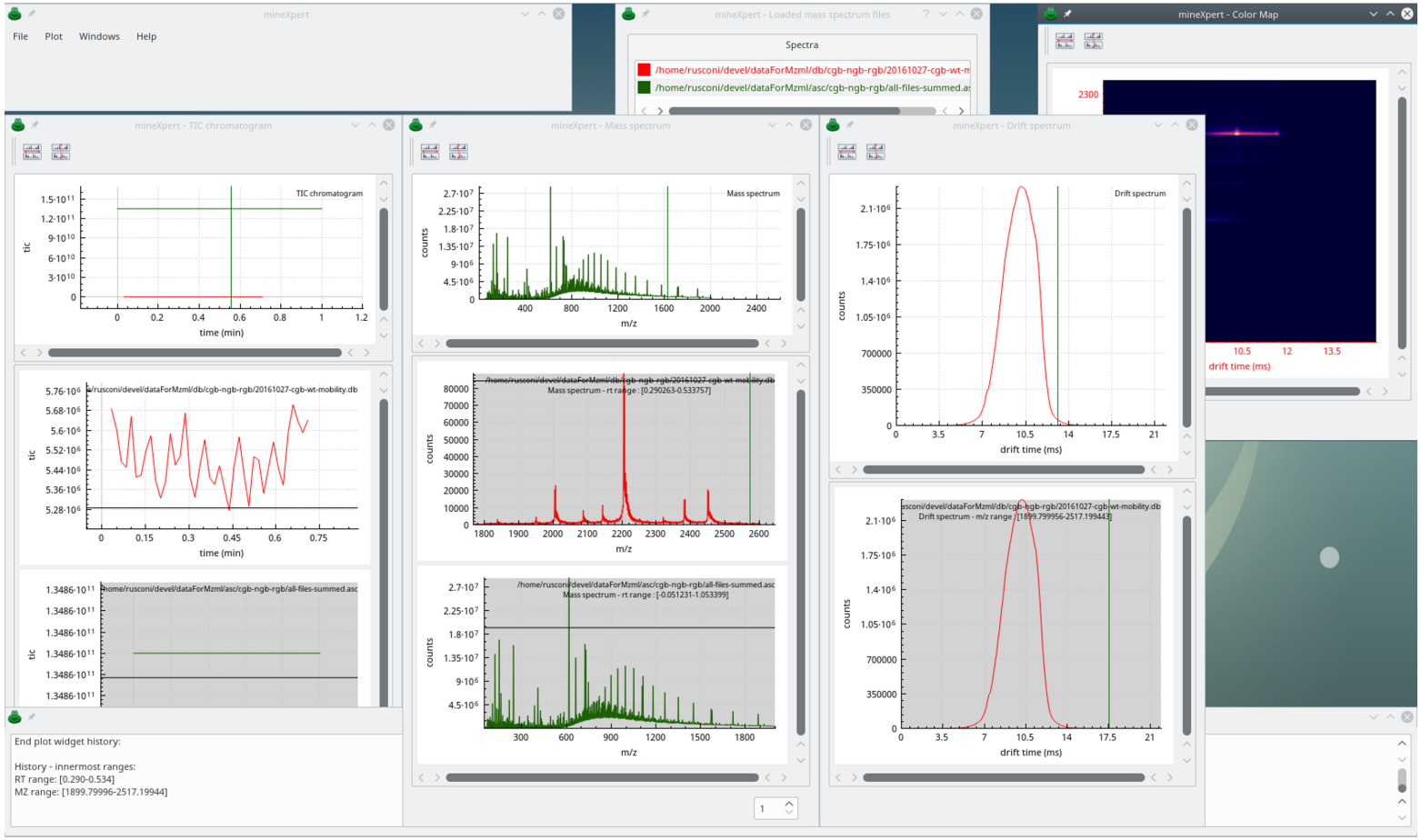


Figure 3.1: **General view of the graphical user interface**

The Main Program Window Menu

The menu bar in the main program window displays a number of menu items, reviewed below:

* *File*
  - *File*⟶*Open full mass spectrum file(s)* Choose the mass spectrum file(s) to load. Note that the "full" descriptor indicates that the user wants to actually load the full data set in memory. This means that she explicitely knows that the system's memory will cope will all the data in the file;
  - *File*⟶*Open streamed mass spectrum file(s)* Choose the mass spectrum file(s) to load. Note that the "streamed" descriptor indicates that the user wants *not* to actually load the full data set in memory. This is typically the case when the data file is so large that its data cannot fit in memory. The program then only "looks" at the data in the file and crafts, piecemeal, the TIC chromatogram and the color map. In this context, any other data integration will be performed by looking into the same mass data file since no data are available in memory;
  - *File*⟶*Mass spectrum from clipboard* creates a mass spectrum from a textual representation of (mz , i) pairs in the same format as described above for the *txt,asc* file format;
  - *File*⟶*Analysis preferences* Define the analysis preferences. The analysis preferences govern how the data about scrutinized mass peaks are recorded to the console or to a selected file, or both.

* *Plot*
  - *Plot*⟶*Clear plots* Clears all the plots currently displayed in the program. The plot items in the Loaded mass spectrum files window are all removed. Note that this releases all the memory that was used by the data. This menu is equivalent to "closing all files";
  - *Windows*

    The menus are self-explanatory, as they explicitely explain which window is to be shown. The *Save workspace* menu records on disk the position and size of all the windows, so that upon reopening the program, the windows all position themselves at the recorded position and size;
  - *Help*

    This menu's items show help about the program itself and also about the Qt libraries that were used to build it. These informations are essential in case the user wants to make a bug report.

The main data windows

This section will succinctly describe the main data windows of *mineXpert*. Each window will be described in greater detail when the features of the program will be described.

The TIC chromatogram window

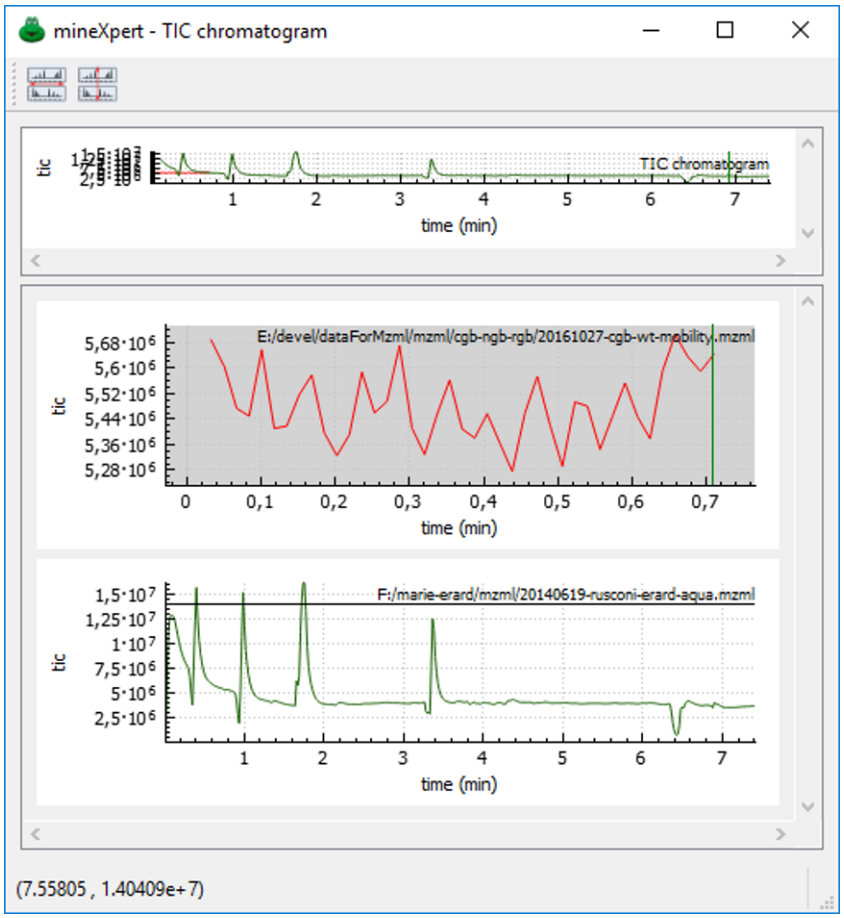


Figure 3.2: **The TIC chromatogram window** This TIC chromatogram window shows multiple chromatograms.

Each time a new mass spectrum file is loaded, its corresponding TIC chromatogram is computed and then displayed in a new plot widget in the TIC chromatrogram window (Figure 3.2). Each new TIC chromatogram plot generated as a result of the loading of a mass spectrometry data file is plotted using a new color. That color encodes the filiation of the whole set of plots that are generated starting from that initial TIC chromatogram plot. For example, a red TIC chromatogram plot that serves as the starting point for a mass spectrum

integration will trigger the creation of a mass spectrum plot widget that will have a red graph in it. Same is true for the color map widget that has its axis and tick labels of the same color as that of the TIC chromatogram plot.

The attention of the reader is drawn on the specific situation corresponding to the loading of mass spectrometric data from a non-profile acquisition data file. For example, when a mass spectrum is opened from a txt,asc,xy text-based format file where the data correspond to a single *spectrum, not a sequence of spectra. In this case, the TIC chromatogram really has a single (rt,i) pair denoting the TIC intensity a the single retention time of that very unique spectrum. The TIC chromatogram window thus artificially created and displayed like shown in Figure 3.3. See the caption of that figure for the explanation.*

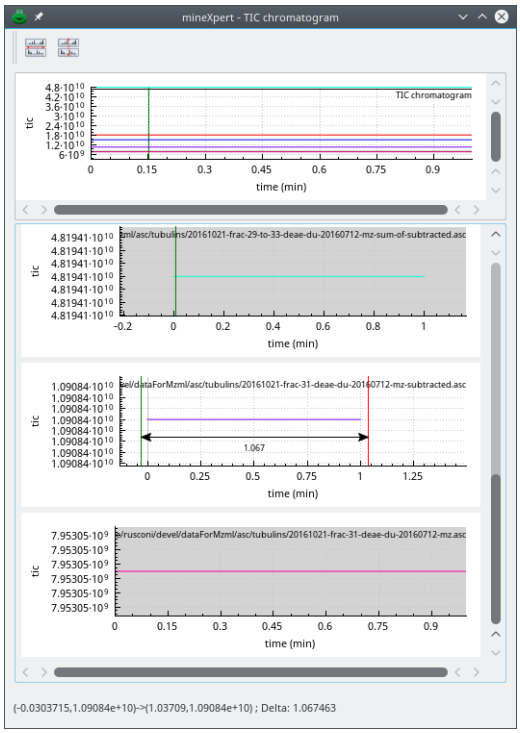


Figure 3.3: **The TIC chromatogram window** This TIC chromatogram window shows the peculiar situation of non-profile acquisitions, for which there is no real TIC chromatogram. The graph is thus artificially created. To perform an integration to a mass spectrum, unzoom the graph and enclose it fully (the two extreme points need to be enclosed) in the integration range as shown for the integration process going on in the middle graph.

The mass spectrum window

The mass spectrum window contains all the plot widgets that display mass spectra that originated in other windows. For example, the user might select a region in a TIC chromatogram and then ask that a mass spectrum integration be computed. In this case, the resulting mass spectrum is displayed in a new plot widget that is located in the mass spectrum window (Figure 3.4).

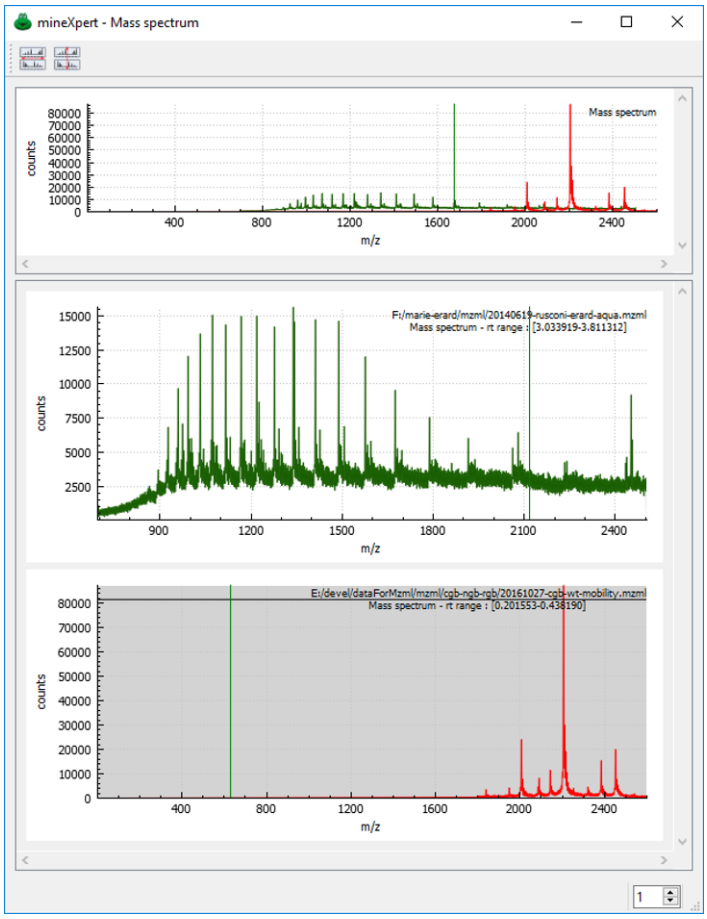


Figure 3.4: **The mass spectrum window** This mass spectrum window shows multiple mass spectra.

The drift spectrum window

As described for the mass spectrum window, the drift spectrum window contains all the plot widgets that display drift spectra (Figure 3.5 on the following page).

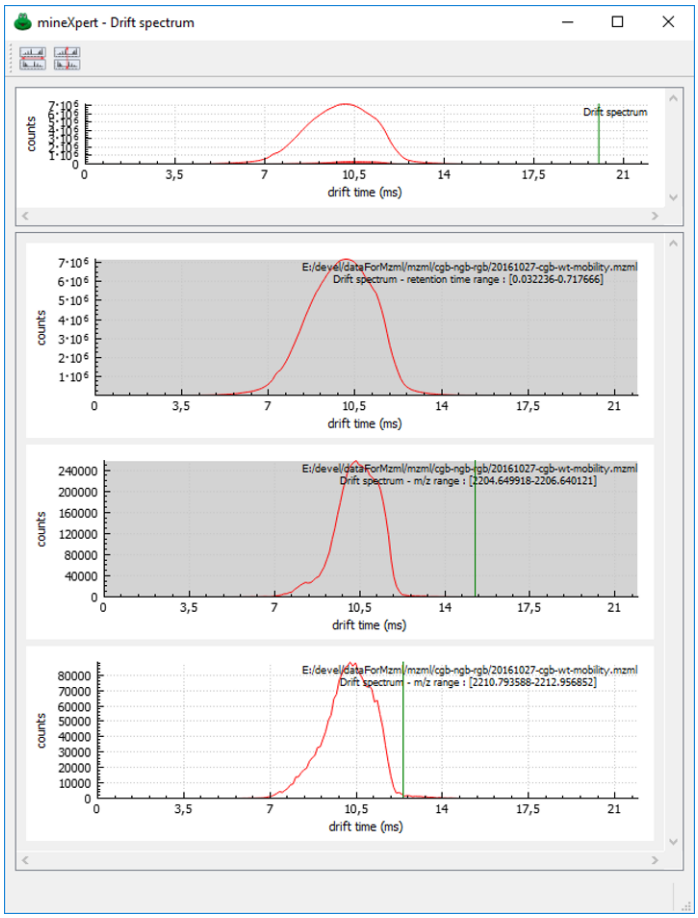


Figure 3.5: **The drift spectrum window** This drift spectrum window shows multiple mass spectra.

The color map window

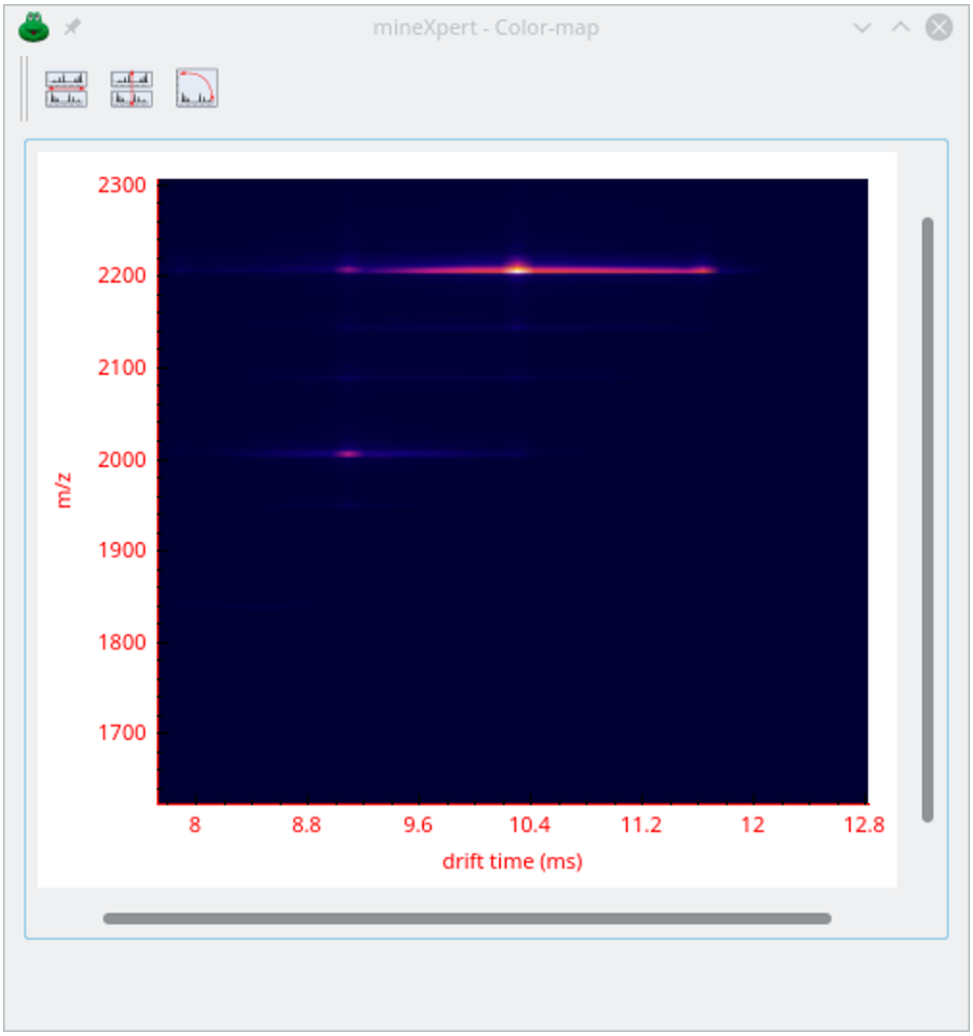


Figure 3.6: **The color map window** The color map window relates mass spectra to drift times.

The color map window displays a color map view of the drift data in the form of m/z *vs* drift time (dt). The intensity of the m/z values is coded in colour. At the time of this writing, only the m/z *vs* dt representation of the drift data are displayed (Figure 3.6).

General structure of the windows containing plot widgets

The TIC chromatogram, mass spectrum and drift spectrum windows are all structured in a similar way. The window is divided vertically in two compartments. The bottom compartment will host all the plot widgets stacked vertically. The top compartment hosts a single plot widget where all the graphs that are displayed unitarily in the lower compartment are shown superimposed.

> *The plot widget that is packed in the top compartment of the window is called the* "multi-graph" *plot widget because it can hold more than one graph. The plot widget(s) that is(are) packed in the bottom compartment of the windows is(are) called* "single-graph" *plot widget(s) because each plot contains only one graph.*

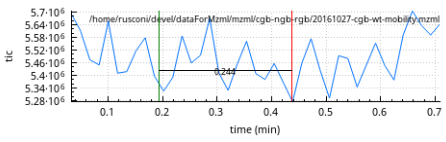
The two vertical compartments of the window are resizable by dragging the sliding horizontal bar that separates them. It is possible to totally occlude one of the compartments by dragging that sliding bar all the way up (or down) to the window side.

The behaviour described above does not apply to the Color map window that has no upper compartment with all the color maps superimposed.

General working of the widgets displaying data

The TIC chromatogram, color map, mass spectrum and drift spectrum windows all contain plot widgets (or color map widgets) that have a general working scheme as to how the data can be visualized. The main visualization operations are succintly described below. The following convention will be used to describe the mouse buttons : 🖱: left mouse button, 🖱: middle mouse button and 🖱: right mouse button.

* Zooming in and zooming out:
  - Zoom in: 🖱-click-drag to draw a selection rectangle. When the mouse button is released, the new plot view contains the data contained in the selection rectangle;
  - Zoom in: 🖱-click-drag along the X-axis over the region to zoom and release the mouse button. The new zoomed view does not automatically scale to full scale in the Y-axis direction. To ensure that the new view automatically scales on the Y-axis, press Shift while releasing the mouse button;



As seen in the figure above, the region defined by the 🖱-click-dragging operation is delimited by green and red markers, respectively at the start and at the end of the selection. The distance between the start and end points is updated along the mouse move operation.

- ✦ Zoom in/out: 🖱-click-drag on the X- or Y-axis to interactively zoom in or out along the selected axis. In this mode, the zoom operates by contracting/extending the data in such a manner that the left/bottom part of the graph (the origin of the graph) is anchored and does not move. When the drag occurs towards larger values on the clicked axis, the view is zoomed in along that axis. Conversely, it is possible to zoom out by dragging the mouse towards lower axis values. When the number of points in the plot is so large that the zoom operation is sluggish, pressing Ctrl will fluidify the zoom operation;
- ✦ Zoom in/out: The 🖱-wheel-rotation can be used to zoom in or out the whole plot on both the X- and Y-axis simultaneously. Note that the position of the mouse cursor when the wheel is rolled defines the new view of the plot. Practice allows to make that zooming in/out mode very powerful.
- ✦ Zoom out: To reset the zoom along one axis, 🖱-double-click that axis. In this case, only the clicked axis will be full-scale, the other axis remains unchanged. To reset the zoom such that the full scale is calculated on the data set displayed after the zoom, maintain the Shift key pressed when double-clicking. To reset the zoom on both axes in one go, 🖱-double-click one of the axes maintaining the Ctrl key pressed;

✱ Panning:

- ✦ 🖱-click-drag on one of the axes to pan the plot view along that axis;

✱ History:

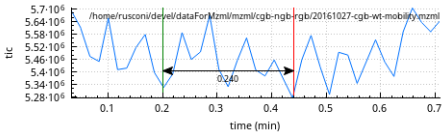
- ✦ Each time a new zoomed in/out view of the plot is triggered, a history element is stored in the plot widget. To back-replay the various steps of the zoom in/out operations in sequence, from pre-last to first, hit the Backspace key. The only exception to this mechanics is when the plot view is panned or when the mouse wheel is used.

The tool bar located at the top of the windows described above contains two button that allow to lock the x axis (the button icon has the horizontal red line) and/or the y axis (red line is vertical) range throughout all the graphs displayed in the window. This of great use when the user wants to compare a number of graphs that have been obtained on comparable samples. The movements and zooming-in or zooming-out operations in one graph are then synchronized to all the other graphs. The third button performs a transpose operation. When the color map is initially created, the horizontal axis (the keys of the map) is the drift time axis and the vertical axis (the values of the map) is the m/z axis. The transpose operation switches the representation of the map such that the axes are inverted.

Data integrations featured by *mineXpert*

Analyzing mass spectrometric data (with or without drift data) usually involves performing various data integrations in sequence. We saw earlier that the first data that are plotted upon loading a mass spectrometry data file are the TIC chromatograms along with (if applicable) the m/z *vs* dt color maps. These two graphed data sets are the starting points for the mass spectrometric data mining, that may involve the following integration operations:

* **TIC chromatogram to mass spectrum** This kind of operation is triggered upon 🖱-click-dragging the mouse over the region of interest and maintaining the [s] key pressed. *mineXpert* integrates all the spectra that have been acquired at all the retention times between the start and the end of the selected region. A new mass spectrum is then plotted in a new plot widget in the mass spectrum window;



  As seen on the figure above, the region defined by the 🖱-click-dragging operation is delimited by arrows, a green marker at the start and a red marker at the end.

* **TIC chromatogram to drift spectrum** This kind of operation is similar to the one described above, unless the [d] key must be pressed. As above, a new drift spectrum is appended to the drift spectrum window.

* **Color map to mass spectrum** This operation involves 🖱-selecting a rectangular region of interest on the color map and by maintaining the [s] key pressed. A new mass spectrum is then plotted in a new plot widget in the mass spectrum window.

* **Color map to drift spectrum** Same as above, but with the [s] key pressed. As above, a new drift spectrum is appended to the drift spectrum window.

* The same mechanics is at work in the other plot widget windows. For example, to trigger the integration of a mass spectrum starting from a drift spectrum, simply drag the mouse over the drift spectrum and maintain the [s] key pressed. Rule of thumb: when a mass spectrum is to be generated, use the [s] key, when a drift spectrum is to be generated, use the [d] key and, finally, when a TIC chromatogram is to be generated, use the [r] key.

* One of the most interesting features for detailed mass data mining is the integration to a TIC intensity. That integration can be triggered from any

of the data window (any plot widget in any of these windows, that is). This integration will be discussed later.

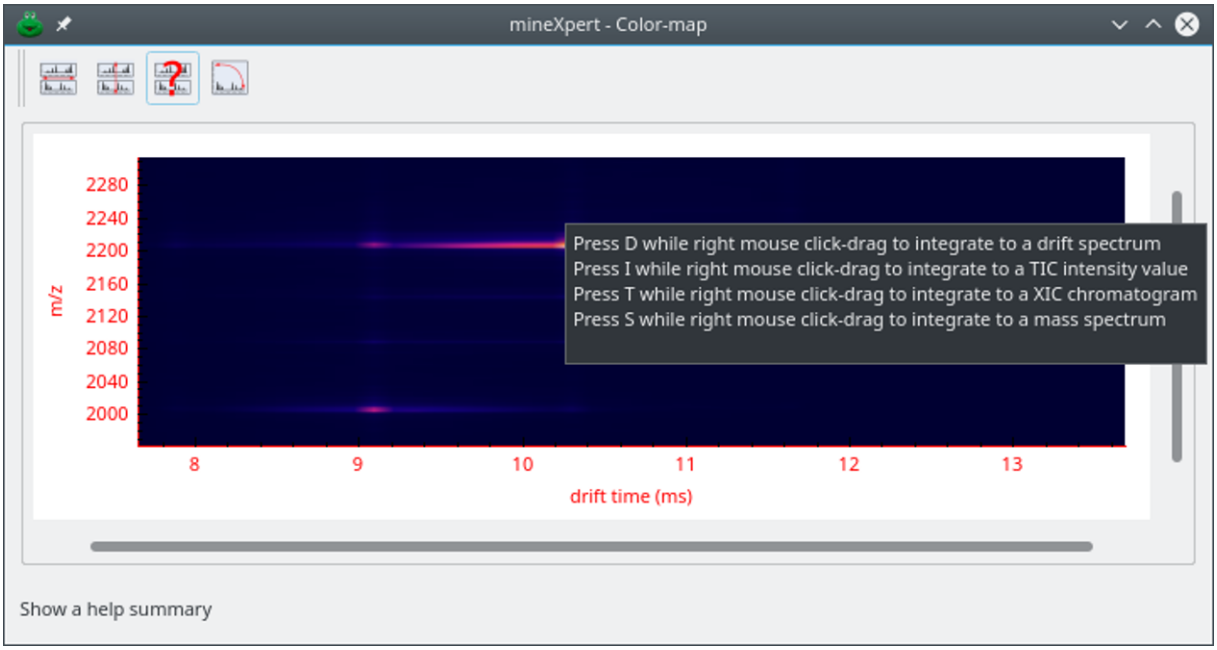


Figure 3.7: **The tool bar and its buttons** The '?' button shows a tool tip helping the user to select the proper keyboard/mouse combination to perform a given integration task.

Figure 3.7 shows the various buttons of the characteristic plot window. The button with the '?' character will show a tool tip describing the various keyboard/mouse combinations to use to trigger the various data combinations described above.

The "filiation" of the plots is maintained using identifying colors. However, it is necessary to be able to maintain a "history" of the way any given plot has derived from the initial TIC chromatogram/Color map plot. This history is shown in a small widget that shows up when the O key is pressed while the mouse cursor hovers over the widget at hand. The history widget is shown in Figure 3.8 on the next page.

Figure 3.8: **The plot filiation history widget** Each plot has a filiation history that can be displayed by keying O.

This figure shows the filiation history of plots. By necessity, the first history item is the [File→RT] element that indicates that the graph originates from loading a file and computing the TIC chromatogram. The range is indicated

(in min). If the user makes an integration to a mass spectrum of the TIC chromatogram region [0.2–0.5], the mass spectrum plot will have the history shown as the second screen dump from the top of the figure. That [RT→MZ] integration is documented as having been performed on the [0.2–0.5] retention time range. If the user now performs an integration from the mass spectrum to a drift spectrum, the filiation history displayed for the drift spectrum shows the [MZ→DT] element, that specifies that the integration was performed for a mass spectral interval [2204.391–2206.843]. It might be very useful to back-integrate the drift spectrum to a mass spectrum to ascertain which m/z values are responsible for a given drift peak or shoulder. This is what has been done, with the fourth screen dump of the figure: that new mass spectrum is the result of an integration in the drift spectrum in the [8.60–10.12] range. Finally, it might be of interest to know where and how the m/z values shown in that mass spectrum originate from in the whole mass spec data set: that is shown in the last screen dump of the figure, where the total ion current extraction is shown with the last history element [MZ→RT] with range [2204-2207].

As can be seen in the various filiation history widgets of Figure 3.8 on the facing page, there is always a History - innermost ranges section that lists the three ranges for RT, MZ and DT. What is that "innermost range" concept? The idea is that, at any given time, the user might desire to know what it the smallest range that a given plot is originating from. In other words, when looking at a plot the user might ask –*"What is the minimum common range that this plot is originating from?"* This is illustrated in the figure with the RT range: while the mass data file had an initial RT range [0.033–0.7], all the remaining plots were created by integrating data for a retention time range [0.2–0.5]. That RT range is the "common smallest range" used in the whole mass data mining. Of course, such common smallest ranges might apply not only for TIC chromatogram plots (or retention time, RT) but for MZ (mass spectral) or DT (drift spectral) plots.

Detailed description of the integration calculations

Depending on the integration that is triggered in the various data plot/map widgets, the computations vary significantly. This section will describe the general computation algorithms in such a manner that the *mineXpert* user can grasp what is actually going on in the guts of the software.

Integrations originating in the TIC chromatogram

When the user integrates data starting from a TIC chromatogram, two calculations may happen:

* Integration to a mass spectrum;
* Integration to a drift spectrum;
* Integration to a TIC intensity value;

Integrating from a TIC chromatogram to a mass spectrum

The integration occurs for a given retention time range in the TIC chromatogram. If we consider an integration range [0–15] min, this is what would occur in the guts of *mineXpert*:

- First of all, create a new mass spectrum (let's call it newMs);
- Extract from the mass spectrometry data all the spectra that have their internal rt value (retention time) contained in the [0–15] min interval. The list of extracted mass spectra (let's call that list msL) is then processed as follows:
- Iterate in msL and for each iterated mass spectrum (ms):
  - Iterate in all the (mz , i) pairs of ms and for each one do:
    - Check if the m/z value was already found in any of the previous mass spectra, that is, if a (mz , i) pair in newMs has that m/z value. If:
      - the m/z was not found, copy the (mz , i) pair in newMs;
      - else if the m/z value was already encountered in previously iterated mass spectra, increment the intensity of the corresponding (mz , i) pair of newMs by the value of the iterated (mz , i) pair. This is where the combination of mass spectra is at work.

At the end of this process, newMs will correspond to the summation of all the spectra contained in the msL list. The newMs mass spectrum is then plotted in the mass spectrum window as a new plot. The color of the newMs plot is the same as the color of the initial TIC chromatogram plot.

Integrating from a TIC chromatogram to a drift spectrum

The integration occurs for a given retention time range in the TIC chromatogram. If we consider an integration range [0–15] min, this is what would occur in the guts of *mineXpert*:

- First of all, create a <dt,tic> map to store all the drift time values encountered below, along with the cumulated total ion current intensity value of the spectra acquired at the corresponding dt drift time.
- Extract from the mass spectrometry data all the spectra that have their internal rt value (retention time) contained in the [0–15] min interval. The list of extracted mass spectra (msL) is then processed as follows:
- Iterate in msL and for each iterated mass spectrum (ms):
  - Get the dt at which ms was acquired;

- Calculate the total ion current (tic) for ms;
- In the <dt,tic> map, check if the dt value was already found. If:
  - the dt was not already found, create one (dt,tic) pair and insert it in the map;
  - else if the dt was already encountered in previously iterated mass spectra, increment the tic value of the corresponding (dt,tic) pair in the map by the tic value calculated above for ms.

At the end of this process, the <dt,tic> map will correspond to the drift spectrum. That spectrum is then plotted in the drift spectrum window as a new plot with the same color as that of the initial TIC chromatogram plot.

Each new mass spectrum is plotted with the same color as the color of the TIC chromatogram that gave rise to it. If multiple mass spectra have been obtained by integration from the same TIC chromatogram, they all will have the same color. This color will allow the traces to be identified in the upper part of this window, where all the spectra are displayed overlaid.

Integrating from a TIC chromatogram to a TIC intensity value

This integration occurs when the user 🖱-selects a range in the TIC chromatogram plot while pressing the [I] key. The integration is performed by looking into the mass data for (mz , i) pairs that match the current integration history of the current data plot and sums all the intensities to yield a final TIC intensity value. This value is printed in the status bar of the window.

> *Worthy of note is the fact that this kind of integration can be performed in the exact same way in the various data plots (mass spectrum, drift spectrum, mz=f(dt) color map).*

Chained Integrations

The user, in the process of mining the data, will inevitably chain integrations to pinpoint a specific feature of interest. For example, let's say that the user performs the following chained integrations (see Figure 3.9 on page 33):

1. From a TIC chromatogram that spans [0–1] min, integrate to a mass spectrum the [0.3–0.6] range;
2. From the mass spectrum, that spans a m/z range of [500–5000], integrate to a drift spectrum the [2202.4–2204.4] range;
3. From the drift spectrum that spans [0–22] ms, where two unresolved peaks are visible, integrate back to a mass spectrum the drift region spanning [10.29–12] ms and then the drift region spanning [12–14] ms.

What happens during all these integrations? In the first integration, the whole m/z range of the mass spectrometric data is considered and the data filtering is only based on the retention time of the spectra that were acquired.

Only those spectra, acquired at retention times between 0.3 and 0.6 min are considered for the combination that will lead to the production of a mass spectrum. In the second integration, two elements are to be taken into account for the filtering of the mass data: the fact that we only care of spectra acquired in the [0.3–0.6] retention time range *and* the fact that we are only interested in the mobility of ions whose m/z value is contained in the [2202.4–2204.4] range. The mobility spectrum show a broad peak with a poorly resolved shoulder. The user wants to get the m/z values that are responsible for these unresolved peaks. She integrates to a mass spectrum the drift region [10.29–12] ms. In this integration the initial mass spectrometric data are filtered according to all the criteria mentioned above: retention time range *and* m/z range *and* drift time range. The same applies for the other drift spectrum to mass spectrum integration starting from drift time range [12–14] ms. As shown in the last two mass spectra, the data is indeed contained in the m/z [2202–2204] range. Incidentally, it appears from the analysis that the same ions have different drift times, since both drift spectrum shoulders back-integrate to very similar mass spectral patterns.

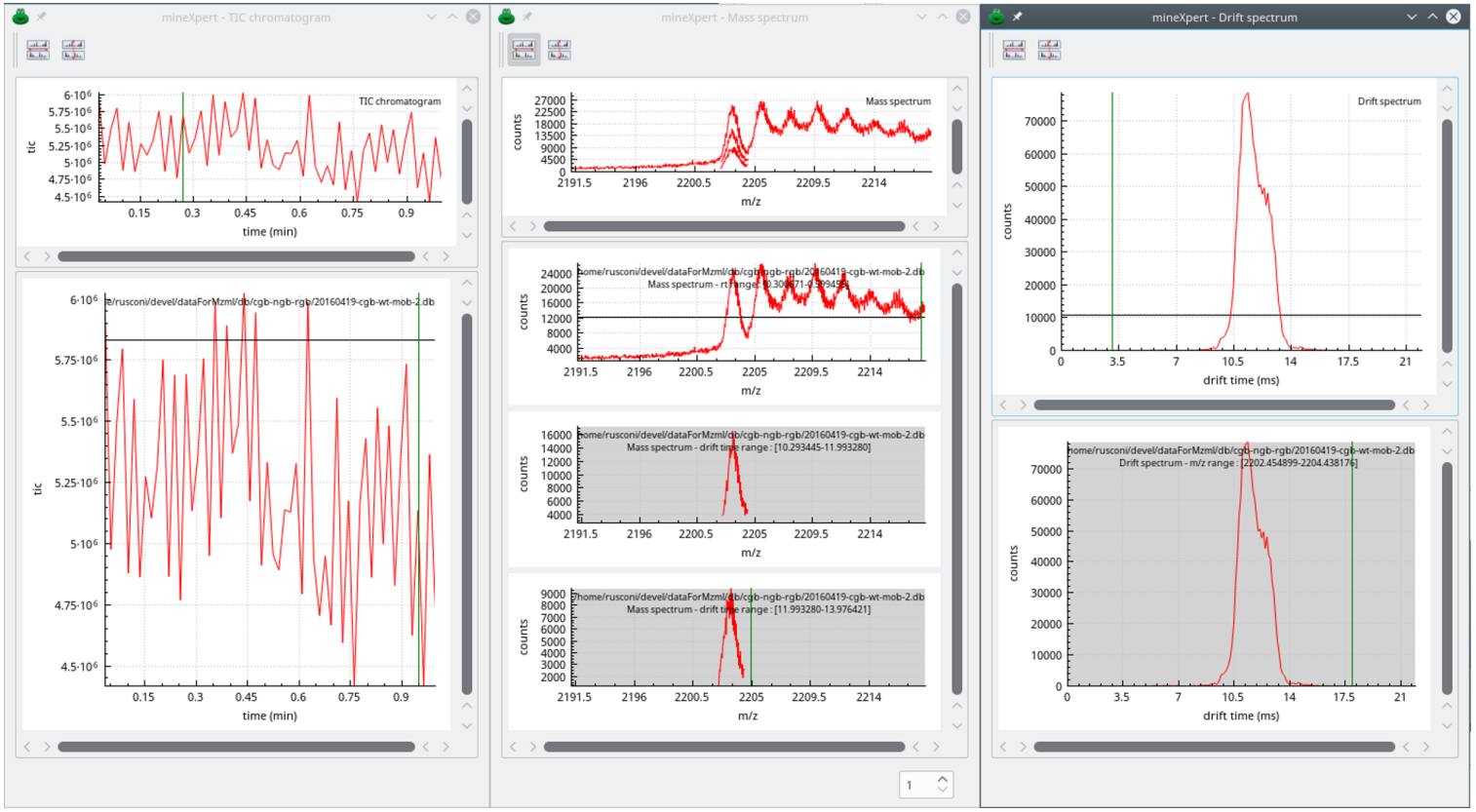


Figure 3.9: **Example of chained integrations**

Mass spectral feature analysis

When analysing a mass spectrum, two major deconvolutions are performed to get back to the mass of the analyte while reading m/z values: the charge state family-based deconvolution and the monoisotopic cluster-based deconvolution. In the following sections, both deconvolutions are described.

Mass spectral deconvolution based on charge state envelope mass peaks

In this kind of deconvolution, at the present time, the software assumes that the ionization agent is the proton and that the ionization is positive.

The deconvolution is based on the determination of the distance between consecutive (or not) peaks of a given charge state envelope. When the user 🖱-click-drags the cursor from one peak to another, the program tries to calculate if the distance between two peaks matches a charge difference of 1 (or more). If so, it computes the molecular (Mr) mass of the analyte whose mass peak is located under the cursor. Figure 3.10 shows that precise state for two *consecutive* peaks of a charge state envelope.

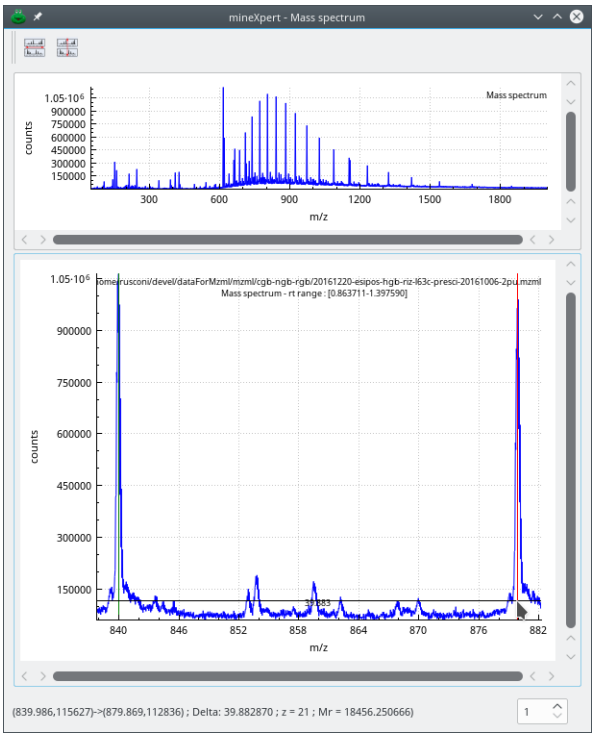


Figure 3.10: **Charge state envelope-based mass deconvolution** Approach using two *consecutive* mass peaks.

The status bar of the window documents the current inter-peak distance

measurement operation that is performed by 🖱-click-drag of the cursor starting at the left peak towards the right peak. The start peak is marked with a green marker and the end peak is marked with a red marker. Start and end positions are documented in the form (m/z start, i)→(m/z end, i). Then, the delta, that is, the distance between both positions is provided. When the end position matches a theoretically expected distance corresponding to a charge difference of 1, then the charge z of the peak under the cursor is provided and the molecular mass (Mr) is provided for the analyte whose peak is under the cursor.

It might happen that two *consecutive* peaks of the charge state envelope are not of a good shape enough to point and click precisely in the center of the peaks. In that case, the software allows indicating the number of intervals that run between two 🖱-click-drag-connected peaks. This is illustrated in Figure 3.11. The user knew that she had to measure the distance between two peaks that were separated by two intervals. She thus incremented the interval value in the status bar to *2* and performed the measurement. The Mr value that is displayed is different than the previous one because without enlarging the window, it is more difficult to click right at the center of the gaussian shape of each peak. Theoretically, the Mr values should be identical, and actually are when the measurements are performed cleanly in widely-laid mass spectra.

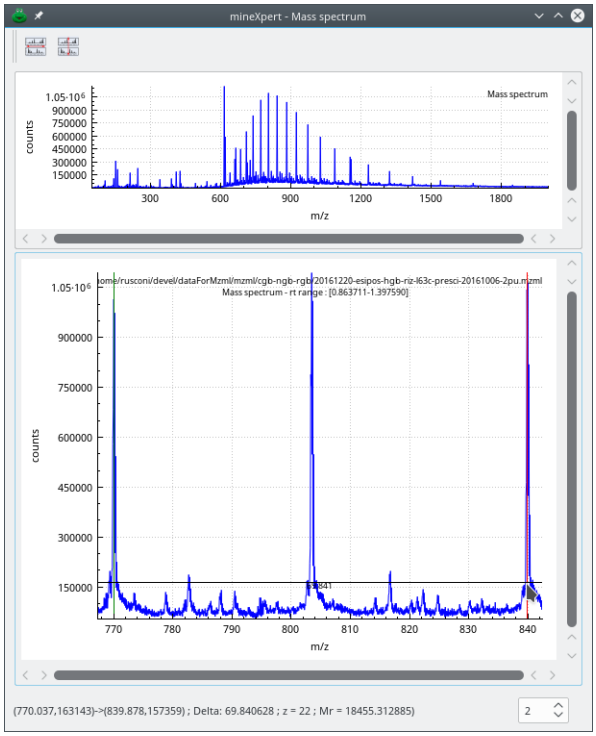


Figure 3.11: **Charge state envelope-based mass deconvolution** Approach using two *non*-consecutive mass peaks. Note the *2* interval value in the status bar of the window.

Note that the 🖱-click-draggging direction (left→right or right→left) has an impact on the value of the charge (z) that is obtained, since that charge value

is relative to the peak *under* the cursor at the moment of the deconvolution. Conversely, the mouse-dragging direction has no effect on the Mr (molecular mass) of the analyte obtained as a result of the deconvolution process.

Mass spectral deconvolution based on isotopic cluster peaks

In this kind of deconvolution, the user 🖱-click-drags the cursor between the first two peaks (when possible) of the isotopic cluster. The charge state of the ion is $\frac{1}{\Delta m/z}$, with $\Delta m/z$ the distance between the two consecutive peaks. Figure 3.12 shows that deconvolution process at work.

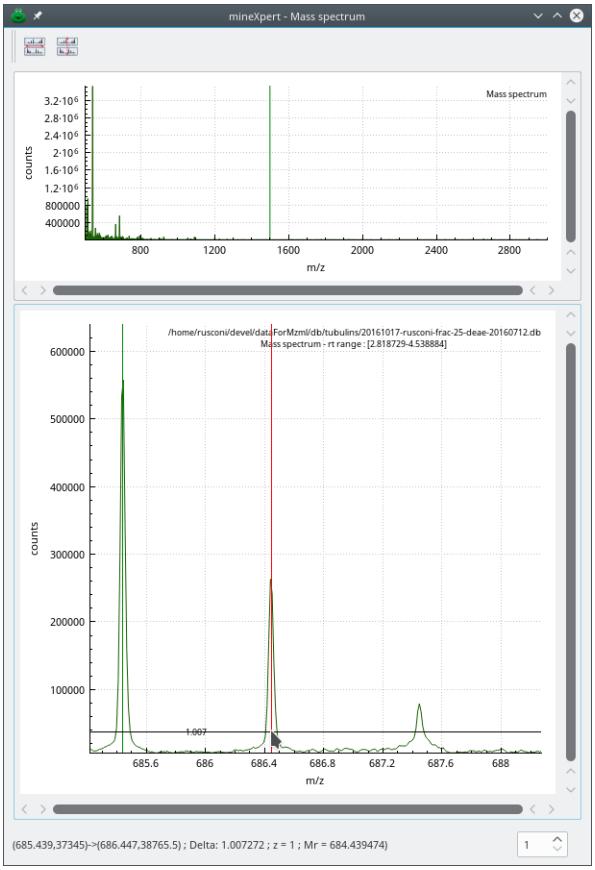


Figure 3.12: **Isotopic cluster-based mass deconvolution**

Note that the direction of 🖱-click-dragging (left→right or right→left) does not have any effect on the Mr value that is obtained as a result of the deconvolution process.

Recording the data mining work

When doing mass analysis work it is often desirable to store the painstakingly manually picked m/z or Mr values for later use. *mineXpert* provides a number of solutions to record the data mining work.

Simplest data record: in-console feature labeling

The simplest way to record any graph feature is to point that feature with the mouse and press the [l] ('el') key. That key shortcut prints to the console window the coordinates of the current mouse cursor location. To be able to trace back the graph source of that (x,y) pair, the text is printed in the console using the same color as the graph whence the labelling action came. The console is actually a rich text format editor in which it is possible to edit the text contents so as to copy/paste them in the lab-book or an email to a colleague, for example. This is shown in Figure 3.13. The label operation described here does not require any previous integration operation. This is in contrast to the requirements of the mass spectral data analysis recording described below.

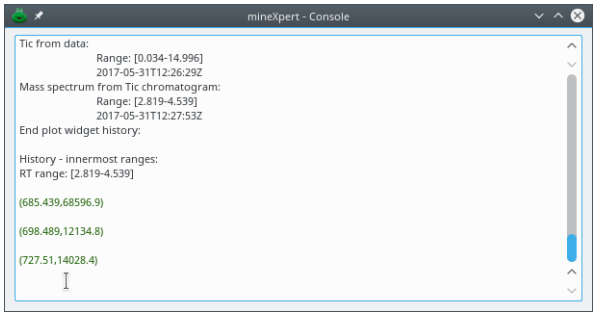


Figure 3.13: **Recording the peak feature coordinates to the console** The text color helps to identify the graph being analyzed.

> *The label recording process works without ambiguity when the cursor is located in the single-graph plot widgets. However, when the cursor is located in the multi-graph plot widget (top part of the window displaying TIC chromatograms, mass spectra or drift spectra) then, only the graph(s) currently selected in the* Loaded mass spectrum files *window is(are) concerned by the label operation.*

Console-/Clipboard-/File-Based Data Analysis Recording

In order to record the innumerous analysis steps that make a data mining session, the *File⟶Analysis preferences* might be called to display the window shown in Figure 3.14. In that window, the user can select the destination of the data

analysis recording system: console, file or both. When selecting file recording, the user might specify if the recording whould overwrite any preexisting file or, instead, append to that file. Depending on the kind of graph where data mining occurs, the format of the data to be recorded will change. For example, it makes no sense to record the charge z when mining data in the Drift spectrum window. This is why the text format of the data export might be defined for the three kinds of graphs: TIC chromatogram, mass spectrum or drift spectrum.

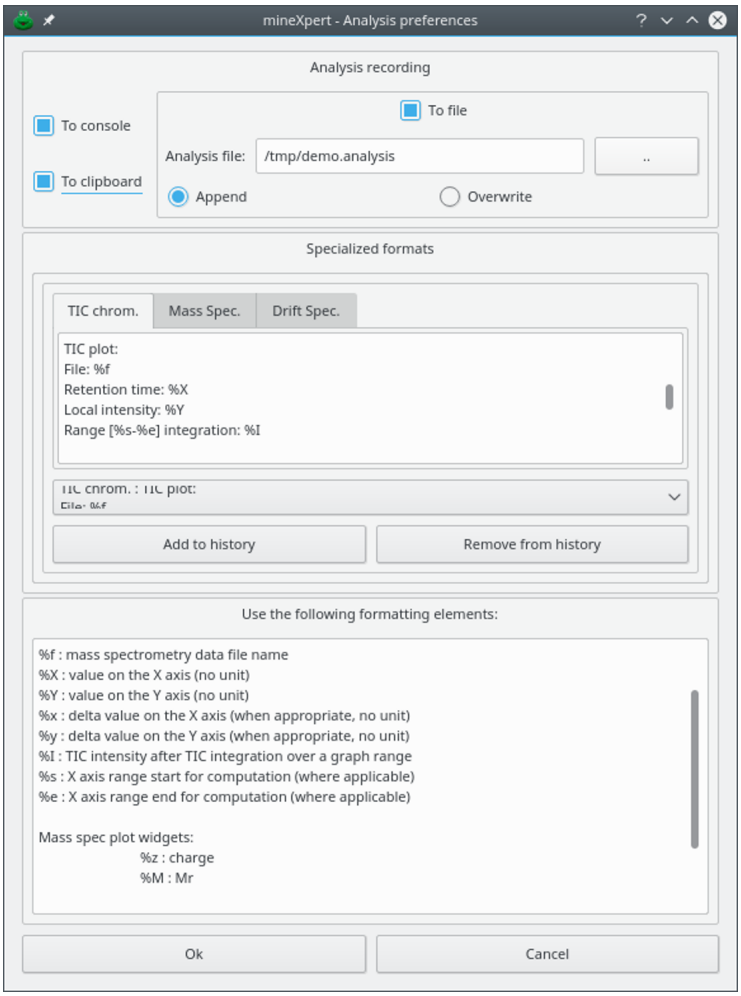


Figure 3.14: **Setting-up of the recording of the data analysis** It is possible to configure the recording system to record to either the console, the clipboard, a file (in append mode or in overwrite mode) or any combination of the three. The format of the string is defined using special characters (see text) and might be defined specifically for the three main graphs: TIC chromatogram, mass spectrum and drift spectrum.

The format used to define the text string to be stored on console and/or in file can contain particular tokens as described below:

* %f : mass spectrometry data file name
* %X : value on the X axis of the graph (no unit)
* %Y : value on the Y axis of the graph (no unit)
* %x : delta value on the X axis (when appropriate, no unit)
* %y : delta value on the Y axis (when appropriate, no unit)
* %I : TIC intensity after TIC integration over a graph range
* %s : X axis range start for computation (where applicable)
* %e : X axis range end for computation (where applicable)

For mass spec plot widgets:

* %z : charge
* %M : Mr

It is important to keep in mind that the %z and %M format strings can only work if the user is actually analyzing a mass spectrum and if the user has effectively *performed a deconvolution operation that has allowed computing these two values. If the values are not available, the program shows nan ("not a number") in the textual output when the space bar is hit (see below).*

In the drift spectrum window, the data recording processes data matching the cursor position at the last 🖱*-single-click. The program tries to define the intensity by looking at the graph ordinate (y axis) matching the nearest abscissa point (x axis) to the last* 🖱*-clicked location.*

Also, as stated above for the simple labelling of cursor location points, the recording of data analysis steps work both in the multi-graph plot widgets (those at the top of the plot windows) and in the single-graph plot widgets (those at the bottom of the windows). When doing data analysis in the top multi-graph widget, it is necessary to select the traces to be analyzed in the Loaded mass spectrum files *window, otherwise no data will be recorded. This is of course not necessary when working in bottom plot widgets because in that case there is no ambiguity on what data to record.*

It is possible to store the format strings in a drop down box for later reuse. Simply click onto the Add to history button while having the format text displayed in the text editor and it will be appended to the drop-down list. The list gets stored when the dialog window is closed and will be filled-up again when the program is restarted.

As an example, if the user defined the following format string for a mass spectrum graph:

```
Mass spec. :
mz = (%X, %Y) z = %z
filename = %f
date = 20161021
session = 20161021
mslevel = 1 msion = esi msanal = tof
chrom = DEAE fraction = 25
seq =  pos =     oxlevel = 0 pos =
intensity =
comment =
```

then, a resulting data mining stanza that would be recorded will look like this:

```
Mass Spec. :
mz = (1051.8, 50863) z = 1
filename = 20161017-rusconi-frac-25-deae-20160712.db
date = 20161021
session = 20161021
mslevel = 1 msion = esi msanal = tof
chrom = DEAE fraction = 25
seq =  pos =     oxlevel = 0 pos =
intensity =
comment =
```

Interestingly, the user can define any kind of format, leaving fields available for later filling-in. This feature if of immense value when the analysis file is used later to fill-in a database for easy storage and interrogation of the mining findings.

At this point, it would be useful to have the file opened in an editor and at each new stanza edit the comment field if something needs to be commented like the shape/intensity of a mass peak, for example.

Note that the program closes the file each time a new stanza has been written. This makes it possible to edit that file safely in between each stanza record. Remember to force the editor to reload the file from disk after each stanza recording.

When the recording involves sending the analysis data to the console, the data are sent to it as text colored the same as the spectrum that was under scrutiny.

When the mouse cursor has been placed at the proper location (with or without  -click-dragging, depending on the situation) on the graph, the user hits the space bar and the data analysis stanza is recorded to the selected destination(s): console, clipboard, file.

4

Converting *mzML* to *SQLite3*

The *mzML* format is very verbose and parsing it causes a notable delay during loading of mass spectrometry files. *mineXpert* allows one to convert *mzML* files to a private open file format based on the `SQLite3` database software.

Using the *SQLite3* format also allows to slice very large data files into smaller files on the basis of user-selected criteria (see *mineXpert* manual).

minexpert can be used in a console without needing to open any window. To show a detailed help, type the following:

```
minexpert --help
```

Use the following parameters (or flags) to perform a data file conversion:

```
minexpert -x -o <db file name> <mzML file name>
```

For example, to convert file `test-file.mzML` into `test-file.db`, the command line would be:

```
minexpert -x -o test-file.db test-file.mzml
```

Alternatively, the `-o` flag can specify a directory, in which case the new file name is crafted from the *mzML* file and written into that directory. In that case, the extension of the *mzML* file needs to be either mzML or mzml for the automatic renaming to occur.

Note that *in batch* conversion is possible using this kind of command line:

```
minexpert -x -o /tmp /home/<user>/lab/mzml/*.mzml
```

In this case, automatic file renaming happens and the new *db* files are all stored in the `/tmp` directory.

5

mineXpert JavaScript-based Scripting

When mining data, often, the user finds herself repeating tasks time and time again. Scripting allows to craft text files in which commands are run by the software program one after the other.

mineXpert makes a scripting environment available to the user. That environment is materialized by Figure 5.1 on the following page.

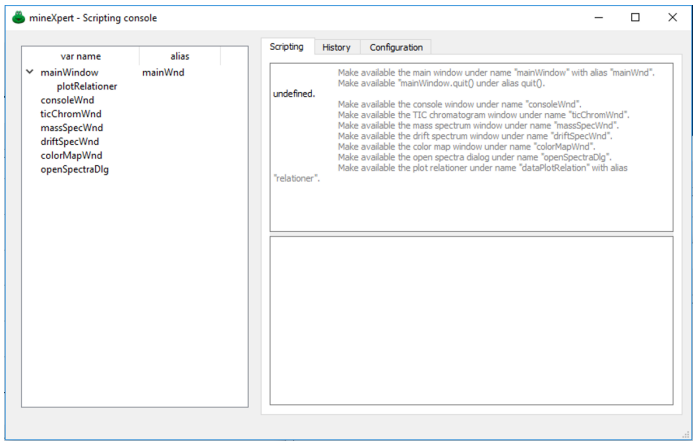


Figure 5.1: **Scripting console, scripting tab**

The left panel is a tree view listing all the objects or function names that are made available to the scripter. The right panel has three tabs. The Scripting tab has two sections; the upper section is where the scripting engine output is recorded; the lower section is the actual console where the user enters the script text.

Using the scripting console

The scripting console is a multiline text edit widget. The user may enter multiple lines in that widget, using the carriage Return key. Once all the lines have been entered to make one or more complete JavaScript statements, the whole set of lines is executed by pressing the Ctrl+Return key combination.

Upon execution of the script statements, any output is recorded in the upper section. When the output obtained from the engine evaluates to true, it is recorded in green color, if it evaluates to undefined, its color is black, and if it evaluates to error, it is colored in red.

Looking at the script engine output section, it is possible to read indented text, colored in grey.

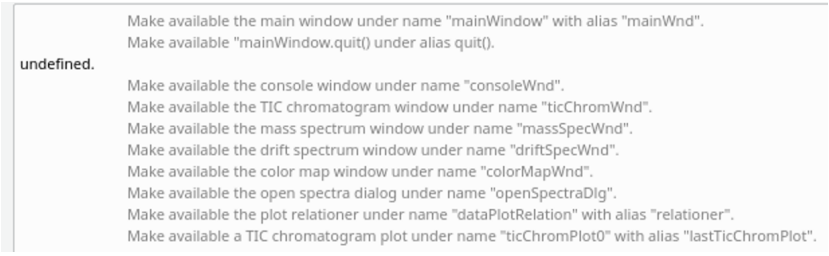


Figure 5.2: **Informative comments to the user**

These are comments that *mineXpert* provides when it runs script statements itself as part of making available to the user globally accessible objects. For

example *Make available the console window under name "consoleWnd"* is an explanatory text for the script statement (that the user does neither enter or see anywhere in the interface because it is run in the guts of *mineXpert*) that creates a globally accessible object for the scripter to use if necessary. The consoleWnd object that is created is listed in the tree view in the left panel of the window.

The available objects list

The left panel of the scripting window lists objects that are made available by *mineXpert* to the scripter. The main objects that *mineXpert* makes available are shown in Figure 5.1 on the preceding page

* mainWindow: the window of the program, where the *File* menu resides;
* consoleWnd: the console window where the various feedback messages to the user are displayed;
* ticChromWnd: the window where all the total ion current chromatograms are shown;
* massSpecWnd: the window displaying all the mass spectra;
* driftSpecWnd: the window displaying all the drift spectra;
* colorMapWnd: the window showing all the mz=f(dt) color maps;
* openSpectraDlg: the window listing all the currently opened mass spectrometry data files;
* scriptingWnd: the where all the scripting occurs.

All the items listed above are systematically available upon launching of the application. For some, they are of no use until a new mass spectrometry file is loaded. Upon opening of a mass spectrometric data file, new objects are created that are made available and listed in this same treeview.

To insert a given object descriptor in the script input editor, double-click the corresponding tree view item.

The history list

Each time a JavaScript statement (single- or multi-line) is executed, that statement gets stored in the History tab that contains a list, as shown in Figure 5.3 on the following page.

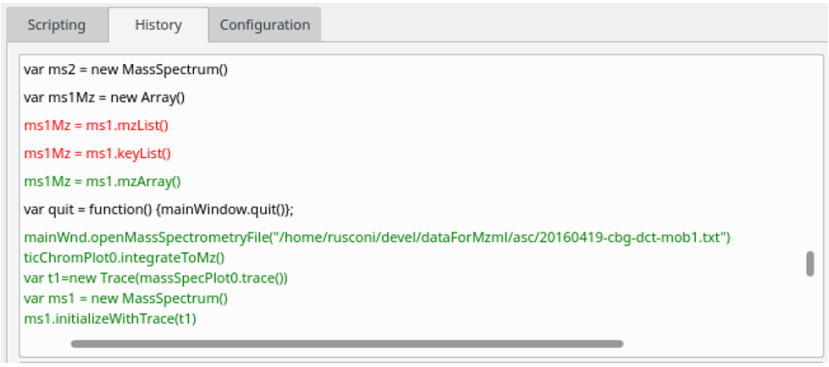


Figure 5.3: **Scripting console, history tab**

The items in the history are reusable by double clicking on them (or selecting them and hitting Return). By doing so, the selected items are copied to the Scripting tab, in the script input widget. It is then only required to hit Ctrl+Return to execute the script line(s).

Exploring the available objects' features

The functional capabilities of the various objects listed in the tree view can be explored by double-clicking the item of interest while maintaining the Ctrl key pressed. The list of functional capabilities is printed in the script output widget, as shown in Figure 5.4.

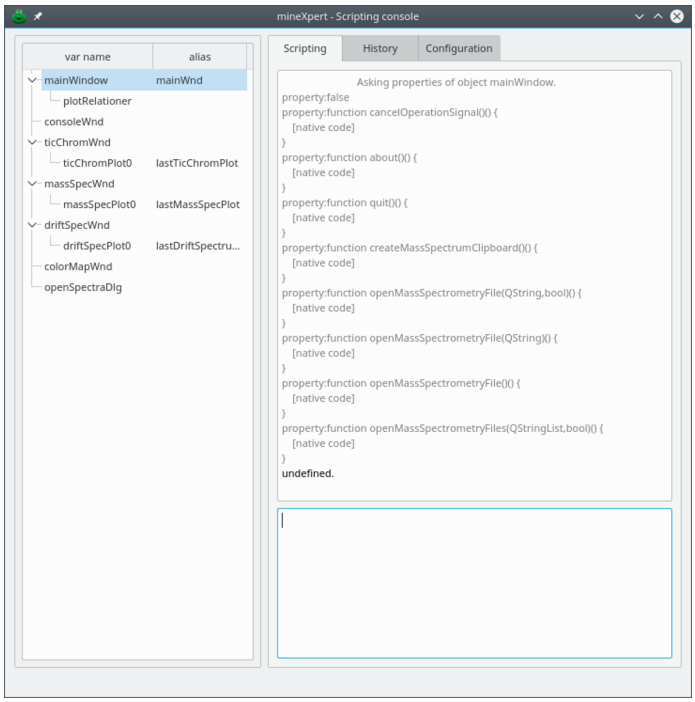


Figure 5.4: **Scripting console, history tab**

As shown in the figure, the mainWindow has a number of member functions that can be called according to the conventional <object>.<function> notation. For example, entering `mainWindow.quit()` in the script input text widget will trigger the `quit()` function of the mainWindow object, effectively quitting the application (the mainWindow is actually the main *mineXpert* window).

As another example, calling `mainWindow.createMassSpectrumClipboard()` will create a new mass spectrum on the basis of data presently in the clipboard, if the data are in the conventional x,y format, with x the m/z and y the intensity.

> *It is essential to understand that not all functions are of use in the scripting context. This is because the listing of the functional capabilities of the objects also comprise functions that are not aimed at the scripting environment but are for internal mineXpert use.*

This is more specifically the case of the plot objects, like ticChromPlot0, massSpecPlot0 or driftSpecPlot0, all listed in the tree view in Figure 5.4. The

creation of these plot objects is discussed in the section below.

Another way to iterate in the JavaScript properties of the objects made available in the JavaScript environment is by using this JavaScript code (using massSpecPlot0 as an example object):

```
for (var prop in massSpecPlot0)
{
print("prop: " + prop);
}
```

This is particularly useful to explore the various functions that are available for any given C++ object that is "exported" to the JavaScript environment (the GlobalObject, really).

Creation of the plot objects as a result of data mining

When new plots are created as a result of the user making data mining operations, new JavaScript objects are created. The JavaScript objects mirror the C++ objects that live in the guts of *mineXpert*. The convention is to name the plot objects made available to the scripting environment using an index, like in *ticChromPlot0*, *ticChromPlot1*... For example, when a new mass spectrometry data file is loaded, *mineXpert* automatically computes the TIC chromatogram and makes it available to the scripting environment under object name *ticChromPlot0*. That name is listed as a child of the *ticChromWnd* object in the tree view of the available objects. This object will remain available as long as the plot lives in the *ticChromWnd*. Note that simulateanously, *lastTicChromPlot* alias is provided to the scripter to capture that newly created object. The scripter willing to store the name of a newly created plot can thus make an assignment like in the following code:

```
mainWindow.openMassSpectrometryFile(‘‘massSpecFile1.mzml’’);

// The last TIC chromatogram plot is made available under the
// temporary name lastTicChromPlot, take advantage to store it
// under a meaningful name for later use

msFile1TicPlot = lastTicChromPlot

// Open a new file so that comparisons can be made
// against the previous one

mainWindow.openMassSpectrometryFile(‘‘massSpecFile2.mzml’’);

// The lastTicChromPlot is made available, take advantage to store
```

```
// it under a meaningful name for later use

msFile2TicPlot = lastTicChromPlot

// Now it is possible to reference the two TIC chromatograms
// with their new names:

msFile1TicPlot.integrateToMz(4.5, 7);
msFile2TicPlot.integrateToMz(4.5, 7);
```

JavaScript-oriented classes

At the time of writing, there are two C++ classes that have a full JavaScript counterpart: Trace and MassSpectrum. In the following sections, a number of use cases are detailed to give the reader a flavor of what is scriptable in *mineXpert*.

Creating a DataPoint object

The C++ class DataPoint has two member data: m_key and m_val. Both are high precision double values. The JavaScript counter part is also named DataPoint and can be initialized in two ways:

```
// create a new DataPoint object, initialized using double values
var dp1 = new DataPoint(123.321,456.654);

// check that this worked fine:
dp1.key;
// --> returns 123.321 (same as dp1["key"])
dp1.val;
// --> returns 456.654 (same as dp1["val"])

dp1.isValid();
// --> returns true

// create a new DataPoint object, initialized using an xy-formatted string
var dp2 = new DataPoint("147.741 258.852");

// check that this worked fine:
dp2.key;
// --> returns 147.741
dp2.val;
// --> returns 258.852

dp2.isValid();
// --> returns true

// create an empty DataPoint object
var dp3 = new DataPoint()

// an empty DataPoint object is invalid
dp3.isValid();
// --> returns false

// a DataPoint can be initialized a posteriori
dp1.initialize(159.951, 357.753)

// or using a string
dp2.initialize("258.852 741.147");
```

The DataPoint class is the main building block Trace objects are made of. A Trace object is nothing but a list of DataPoint instances.

Creating Trace objects

A JavaScript Trace object is nothing but an array of DataPoint objects. One useful application of scripting is to perform a variety of calculations on a given Trace or on a set of Trace_s.

A JavaScript Trace object can be created and initialized in a number of ways.

Creating an empty Trace and initializing it

```
// create a new Trace using an xy-formatted string
var t1 = new Trace("This is the title of the trace object");
t1.title
// --> returns ‘‘This is the title of the trace object’’

t1.initialize("123.321 456.654\n789.987 147.741\n258.852 369.963");
// --> returns 3 // the number of created DataPoint objects

// check that the trace has effectively 3 items:
t1.length
// --> returns 3

// check that the initialization was ok
print("(" + t1[0 ].key + "," + t1[0].val + ")")
// --> returns (123.321,456.654)

// create a new Trace using two arrays of numbers (key list and value list)
t2.initialize([123.321,456.654,789.987],[147.741,258.852,369.963])
// --> returns 3 (the number of created DataPoint objects)

// check actual length
t2.length
// --> returns 3

// check that the initialization went fine
t2[0].key
// --> returns 123.321
t2[0].val
// --> returns 147.741

// get the key and value lists
t2.keyArray()
// --> returns 123.321,456.654,789.987
t2.valArray()
// --> returns 147.741,258.852,369.963

// once a Trace has been initialized, it can undergo interesting actions:

// calculate the sum of all the values:
t2.valSum()
// --> returns 776.556

// a Trace object can combine into itself another Trace object
t2.combine(t1)

// now check if the t2 values have increased accordingly
t2.valArray()
// --> returns 604.395,369.963,258.852,517.704 // correct combination

// now subtract t2 from t2:
t2.subtract(t2)

// now check that subtraction worked fine
t2.valArray()
// --> returns 0,0,0

// it is possible to initialize a trace with limitation to a given key range
var t3 = new Trace()
t3.initializeRange([123.321,456.654,789.987], [147.741,258.852,369.963], 200, 500);

// there should be only one DataPoint in t3
t3.length
// --> returns 1
t3.keyArray();
// --> returns 456.654
```

Creating a Trace object starting from a plot

A Trace object can be easily created by using data already obtained from using *mineXpert*. For example, a Trace object can be made so that it reproduces a plot dispalyed in the graphical user interface. The process is shown below:

```
// load mobility data file
mainWnd.openMassSpectrometryFile(‘‘mobility.mzml’’)

// integrate over the full TIC chromatogram range
// creates the  massSpecPlot0 object

ticChromPlot0.integrateToMz()

// create a new Trace JavaScript object
var t0 = new Trace()

// now initialize t0 with the plot data of ticChromPlot0
t0.initialize(massSpecPlot0.keys(),massSpecPlot0.values());

// check that the initialization actually worked
t0.length // should return the number of DataPoint objects in that Trace

// load a blank data file that acts as a baseline data file
mainWnd.openMassSpectrometryFile(‘‘mobility-blank.mzml’’)

// integrate over the full TIC chromatogram range
// creates the massSpecPlot1 object

ticChromPlot1.integrateToMz()

// create Trace JavaScript object
var t1 = new Trace()

// now initialize t1 with the plot data of ticChromPlot1
t1.initialize(massSpecPlot1.keys(),massSpecPlot1.values());

// check that the initialization actually worked
t1.length // should return the number of DataPoint objects in that Trace

// finally perform some useful task: subtract noise from t0
t0.subtract(trace1)

// Trace object t0 now has the baseline removed.
```

Exporting a Trace object to a file

It is possible at each moment to export a given Trace object (or plot) to a file.

This is possible either at the JavaScript Trace object level:

```
mainWnd.openMassSpectrometryFile("mobility.mzml")
ticChromPlot0.integrateToMz()
var t0 = new Trace()
t0.initialize(massSpecPlot0.keys(),massSpecPlot0.values());
t0.exportToFile("/home/rusconi/demo.xy");
```

Or, at the plot level:

```
mainWnd.openMassSpectrometryFile("mobility.mzml")
ticChromPlot0.integrateToMz()

// note how the export function name has ‘‘Plot’’ in it.
massSpecPlot0.exportPlotToFile("/home/rusconi/demobis.xy");
```

At this point both files demo.xy and demobis.xy are bit-by-bit identical.

Plotting a JavaScript Trace object to a plot wiget window

It is necessary for the data miner to be able to look at her data graphically. Nobody wants to scrutinize data in ASCII files.

Anytime the scripter has a Trace object, she can plot it in the window that makes sense for the data. For example, if a plot derives from mass spectral data, then logically, the new plot should derive for these mass spectral data, as exemplified below:

```
mainWnd.openMassSpectrometryFile("mobility.mzml");
ticChromPlot0.integrateToMz();

var t1 = new Trace();
t1.initialize(massSpecPlot0.keys(), massSpecPlot0.values());

// and now plot that Trace object as a descendant of massSpecPlot0.
// if massSpecPlot0 is removed, then the descendant will be removed also
massSpecPlot0.newPlot(t1);
```

Displaying a new plot using the `<existingPlot Object>.newPlot(<Trace>)` call occurs in the window that matches the type of <existingPlot>. If <existingPlot> is a mass spectrum, then the descendant plot will be displayed in the Mass Spectrum Window.

Any of the plot widget windows can receive a new plot according to the mechanics described above.

Printing a JavaScript Trace object to the console

It is possible to have a look at the numerical data of a Trace object like so:

```
mainWnd.openMassSpectrometryFile("mobility.mzml");
ticChromPlot0.integrateToMz();

var t1 = new Trace();
t1.initialize(massSpecPlot0.keys(), massSpecPlot0.values());

// print the data to the console
t1.asText()

// outputs a series of xy-formatted lines like so:
// 2504.3185084749 1262.0000000000
// 2504.3503266026 1277.0000000000
// ...

// doing this is also possible at the plot widget level
// note the function name change
massSpecPlot0.asXyText()
```

6

Appendices

GNU General Public License Text

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

Terms and Conditions

0. Definitions.

   "This License" refers to version 3 of the GNU General Public License.

   "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

   "The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

   To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

   A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

   The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

   A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

   The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

   The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

   All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

   You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

   Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

   No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

   When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

   You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

   You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

   You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   (a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

   (b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

   (c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

   (d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

   A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

   You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

   (a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

   (b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for

software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

(c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

(d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

(e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a

transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

(a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

(b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

(c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

(d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

(e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

(f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

    An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

    You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

    A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

    A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

    Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you

convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

    Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

    If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

    Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

    THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS

PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

    If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

End of Terms and Conditions

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.>
>
> Copyright (C) <year> <name of author>
>
> This program 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.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

> <program> Copyright (C) <year> <name of author>
> This program comes with ABSOLUTELY NO WARRANTY; for details type `show w`. This is free software, and you are welcome to redistribute it under certain conditions; type `show c` for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see http://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read http://www.gnu.org/philosophy/why-not-lgpl.html.