i2MassChroQ is one of the PAPPSO facility software projects

i2MassChroQ User Manual

Free and Open Source Protein Identification Software

i2MassChroQ 1.0.4

i2MassChroQ User Manual: Free and Open Source Protein Identification Software

by Benoît Valot, Olivier Langella, Thomas Renne, Filippo Rusconi, and Michel Zivy

i2MassChroQ

HTTP://PAPPSO.INRAE.FR/EN/BIOINFO/I2MASSCHROQ/

This book is part of the *i2MassChroQ* project.

The *i2MassChroQ* project is the successor of the Java language-based homonymous project. This project is a full rewrite of the former project in the C++ language, with many new features added.

1.0.4

Revision History

Revision 1.0.0	02 November 2023	Filippo Rusconi

- Added entire section with step-by-step procedure to describe the new *MSstats* interface inside of *i2MassChroQ*.

Revision 0.7.24	28 May 2021	Filippo Rusconi

- Start actually documenting the software with the Preface.

Revision 0.7.24	21 April 2021	Filippo Rusconi

- Very first setting up of the user manual using the mineXpert2 project as a template.

Dedication

To all the admirable people acting in the *"Free Software Movement"* for a better and more ethical computing world

To all the readers who helped with this manual.

Contents

List of Figures

Preface

1 Software feature offerings and intended audience

This manual is about the *i2MassChroQ* protein identification software project.

i2MassChroQ has the following features:

- Load mass spectometry data files in the mzXML or mzML format, thanks to the excellent *libpwiz* library of ProteoWizard[1] fame.

- Configure the way the peptide/mass spectrum matches (PSM) are to be performed;

- Configure the database files to be used (target organism databases and contaminant databases);

- Use the MS/MS data in the file to feed the *X!Tandem* program that produces peptide identification results by matching the measured ion masses with peptide fragments calculated *in silico* on the basis of the databases contents;

- Perform the protein inference step that leads to reliable protein identifications on the basis of the peptide identifications performed by *X!Tandem*

- Display the data obtained at any step in powerful ways in a unified graphical user interface to allow the user to inspect the peptide identifications and also control the way these identifications are used to infer the protein identifications.

- Export the data after the results exploration above in a variety of formats.

- Perform quantitative proteomics on the basis of the results obtained at the previous steps.

- Perform bio-statistical analyses on the quantitative proteomics data obtained at the previous step.

2 Feedback from the users

We are always grateful to any constructive feedback from the users.

The PAPPSO software team might be contacted *via* the following contact page:

HTTP://PAPPSO.INRAE.FR/EN/TRAVAILLER_AVEC_NOUS/CONTACT/ ↗ (search for team members having the "Bioinformatics" specialty mentioned, like Olivier Langella or Filippo Rusconi).

---

1 HTTP://PROTEOWIZARD.SOURCEFORGE.NET/ ↗ .

3 Program and Documentation Availability and License

The programs and all the documentation that are shipped along with the *i2MassChroQ* software suite are available at HTTP://PAPPSO.INRAE.FR/EN/BIOINFO/XTANDEMPIPELINE/ ↗ . Most of the time, a new version is published as source, and as binary install packages for *MS-Windows* (64-bit systems only).

For *GNU/Linux*, binary packages are created locally (see HTTP://PAPPSO.INRAE.FR/EN/BIOINFO/XTANDEM-PIPELINE/DOWNLOAD/ ↗) but are also built in the *Debian*[2] autobuilders and are uploaded to the distribution servers. These packages are available using the system's software management infrastructure (like using the *Debian*'s `apt` command, for example, or the graphical application).

The software and all the documentation are all provided under the Free Software license *GNU General Public License, Version 3, or later, at your option*. For an in-depth study of the *Free Software* philosphy, the reader is kindly urged to visit HTTP://WWW.GNU.ORG/PHILOSOPHY ↗ .

---

2 HTTP://WWW.DEBIAN.ORG/ ↗

1 GENERALITIES

In this chapter, I wish to introduce some general concepts around the *i2MassChroQ* program, the reference to be used to cite the software in publications, the building and installation procedures.

1.1 HISTORY OF THE PROJECT

i2MassChroQ is the successor of the *X!TandemPipeline-Java* project that has seen the following changes along the years:

- Full rewrite of the *X!TandemPipeline-Java* program from Java to C++17. The Java-based software program had been published in Olivier Langella, Benoît Valot, Thierry Balliau, Mélisande Blein-Nicolas, Ludovic Bonhomme, and Michel Zivy (2016) *X!TandemPipeline: A Tool to Manage Sequence Redundancy for Protein Inference and Phosphosite Identification.* in *J. Proteome Res.* 2017, 16, 2, 494–503. https://doi.org/10.1021/acs.jproteome.6b00632.

  ### 💡 TIP

  Before the integrations described below, the product of the rewrite has been called transitorily *X!TandemPipeline++* (or *xtpcpp*). That name might appear in some places while the code/documentation is being revised to change its name to *i2MassChroQ*.

- Integration into the new software of the *MassChroQ* software project that was developed as a standalone C++ software piece. *MassChroQ* is a software project that was developed to perform quantitative proteomics in a variety of modes (label-free or with labelling).

- Unfinalized integration of the *MCQ~R* project that was developed as a standalone project. *MCQ~R* is a GNU R project aimed at performing bio-statistical analyses on the quantification analysis performed by *MassChroQ*.

The *i2MassChroQ* project encompasses three main quantitative proteomics fields of endeavour:

- Database search, peptide identification and protein inference. The database search is actually performed by *X!Tandem* and is started seamlessly by *i2MassChroQ*. Protein grouping is performed by original code in *i2MassChroQ*.

- Quantitative proteomics, mainly based on area-under-the-curve processes (requires the full mass data set to extract ion current chromatograms, XIC). This part was historically performed by the *MassChroQ* software program.

- Bio-statistical analysis of the quantification data. This part was historically performed by the *MCQ~R* GNU R-based package (unpublished software as of yet).

1.2 What does *i2MassChroQ* Stand for?

The *i2MassChroQ* software project aims at providing users with an integrated software solution for quantitative proteomics. As described in detail in another chapter of this book, quantitative proteomics involve a number of steps that can be enumerated in sequence below:

- Search databases to connect MS/MS spectra to peptide sequences. This step is called *identification*;

- Apply logic to reliably identify proteins based on the peptides identified at the previous step. This step is called *inference*;

- Optionally perform quantification of the *i*dentified peptides and *i*nferred proteins. *i2MassChroQ* has area-under-the-curve quantitative proteomics capabilites that are based on precursor peptide ion current extraction from the mass spectrometric data. The extracted ion currents are then plot like chromatograms: intensity as a function of retention time. This analytical process thus somehow involves "*Mass Chro*-matograms" for the *Q*uantification.

From the sequence above, the *i2MassChroQ* name becomes self-explanatory!

Tip

It is however possible (and encouraged) to mentally read *i2MassChroQ* as "*I too MassChroQ !*"

1.3   Transitioning from *X!TandemPipeline* to *i2MassChroQ*

The previous *X!TandemPipeline* version of this software did store configuration data in the local configuration directory and in the `PAPPSO/xtpcpp.conf` file. In order to preserve these configuration data after having transitioned from *X!TandemPipeline* to *i2MassChroQ*, please, rename that configuration file to `PAPPSO/i2masschroq.conf`.

1.4   General concepts and terminologies

This section describes the general concepts at the basis of the analysis of proteomics data that one needs to grok in order to properly assimilate the workings of the *i2MassChroQ* software.

1.4.1   Bottom-up Proteomics or Top-down Proteomics?

Proteomics is a mass spectrometry-based field of endeavour that is aimed at characterizing the "protein complement" of a given genome. The protein complement of a genome is the set of proteins that are expressed at a given instant in the life of a cell, a tissue or an organ, for example. Characterizing that protein complement actually means identifiying the proteins expressed by a given living cell or tissue or organ. Optionally, if feasible, the characterization of post-translational modifications might be desirable.

There are two main variants of protemics: "bottom-up" proteomics and "top-down" proteomics:

- The first variant—bottom-up proteomics—identifies proteins on the basis of the identification of all the peptides obtained by first digesting all the proteins of the sample using an enzyme of known specificity. In this variant, the sample that is injected in the mass spectrometer is the resulting peptide mixture (first resolved by high performance liquid chromatography). The identification of the proteins contained in the initial sample is performed in a number of steps that are actually the focus of *i2MassChroQ*. Indeed the *i2MassChroQ* software is a bottom-up-oriented software program.

- The second variant—top-down proteomics—identifies proteins on the The second variant identifies proteins on the basis of intact proteins directly injected in the mass spectrometer. Of course, it might be necessary to fragment the proteins in the mass spectrometer and to use the fragments to actually identify the protein. However, the fact that the protein is first detected and analyzed as one entity (and not as set of peptides), allows for some very useful discoveries, like the identity and number of post-translational modifications, for example.

Note

At the moment, *i2MassChroQ* does not handle top-down proteomics data: it is a bottom-up proteomics software project.

1.4.2 Typical cycle of a mass spectrometer data acquisition

Once the initial sample, containing all the proteins to identify, has been digested using a protease of known cleavage specificity (trypsin, typically), the peptidic mixture (that might be highly complex) needs to be resolved as much as possible using chromatography. In the vast majority of the proteomics experimental settings, the chromatography setup is connected to the mass spectrometer so that when the gradient is developed, all the peptides are immediately injected "on line" to the mass spectrum ion source.

The mass spectrometer runs an analysis cycle that can be summarized like the following:

- Acquire a full scan mass spectrum of the whole set of ions at a given chromatography retention time. This kind of mass spectrum is called a MS spectrum;

- Enter a loop during which ions having the most intense signal are subjected in turn to collision-induced dissociation (CID), that is, are fragmented by accelerating them against gas molecules in a fragmentation cell. The mass spectra that are collected at each one of these fragmentation acquisitions are called MS/MS spectra because they are obtained after two mass analysis events: the first event is the measurement of the intact peptide ion's m/z value (full scan mass spectrum) and the second event is the measurement of all the obtained fragments' m/z values (MS/MS scan).

Each instrument records all the MS and MS/MS spectra in a raw data format file that is specific of the vendor. Free Software developers cannot know the internal structure of the files. To use the mass spectrometric data, they need to rely on a specific software that performs the conversion from the raw data format to an open data format (mzML). That program is called *msconvert*, from the *ProteoWizard* project.

Note

Mass spectrometrists used to call ions that were analyzed in full scan mass spectra "parent ions". They also used to call fragment ions arising upon fragmentation of a parent ion "daughter ions". This terminology has been deprecated and has been replaced with "precursor ion" and "product ion", respectively. In our document, we thus use the new terminology.

1.4.3 OUTLINE OF AN *i2MassChroQ* WORKING SESSION

i2MassChroQ loads mzXML- and mzML-formatted files and needs for its operations to have accesss to all the MS and MS/MS spectra. Once data files have been loaded, *i2MassChroQ* allows the user to perform the following tasks, that will be detailed in later chapters:

- Configure the *X!Tandem* database searching software (that is, the software, external to *i2MassChroQ* that actually performs the peptide-mass spectrum matches);

- Run the *X!Tandem* software and load its results;

- Display the results to the user in a way that they can be scrutinized and checked. The peptide identification results serve as the basis for another processing step that is integrally performed by *i2MassChroQ*: the "protein inference". That step aims at using the peptide identifications to actually craft a list of proteins identities. The user is provided with various means to control that step in various ways.

- Optionally start the *MassChroQ* module to perform the quantitative proteomics on the identification data checked at the previous step.

- Optionally start the *MCQ~R* module to perform the bio-statistical analysis of the quantitative proteomics data obtained at the previous step.

1.5 CITING THE *i2MassChroQ* SOFTWARE.

The *i2MassChroQ* software is not yet published. A great part of the features found in *i2MassChroQ* have initially been found in the *X!TandemPipeline-Java* software that can be cited using the following reference: Olivier Langella, Benoît Valot, Thierry Balliau, Mélisande Blein-Nicolas, Ludovic Bonhomme, and Michel Zivy (2017) *X!TandemPipeline: A Tool to Manage Sequence Redundancy for Protein Inference and Phosphosite Identification.* in *J. Proteome Res.*, 16, 2, 494–503. DOI: HTTPS://DOI.ORG/10.1021/ACS.JPROTEOME.6B00632 ⬈ .

1.6 INSTALLATION OF THE SOFTWARE

The installation material is available at HTTP://PAPPSO.INRAE.FR/EN/BIOINFO/XTANDEMPIPELINE/DOWN-LOAD/ ⬈ .

1.6.1 INSTALLATION ON MS WINDOWS AND macOS SYSTEMS

The installation of the software is extremely easy on the MS-Windows and macOS platforms. In both cases, the installation programs are standard and require no explanation.

1.6.2   Installation on Debian- and Ubuntu-based systems

The installation on Debian- and Ubuntu-based GNU/Linux platforms is also extremely easy (even more than in the above situations). ; is indeed packaged and released in the official distribution repositories of these distributions and the only command to run to install it is:

$ [1] **`sudo apt install <package_name>`** `RETURN`

In the command above, the typical *package_name* is in the form `i2masschroq` for the program package and `i2masschroq-doc` for the user manual package.

Once the package has been installed the program shows up in the *Science* menu. It can also be launched from the shell using the following command:

$ **`i2masschroq`** `RETURN`

---

[1] The prompt character might be *%* in some shells, like *zsh*.

2 Fundamentals in Bottom-up Proteomics

This chapter is an optional chapter which the reader might be referred to upon reading other part of this manual.

2.1 The Protein Biopolymer: Structure and Chemistry

This section introduces the basics in protein polymer chemistry. The way this topic is going to be covered is admittedly biased towards mass spectrometry and proteins. Moreover, the aim of this chapter is to provide the reader with the specialized words that will later be used to describe and explain the (inner) workings of the *i2Mass-ChroQ* program. This manual is not a "crash course" in biochemistry.

2.1.1 Protein Biosynthesis

Proteins are made of amino acids. There are twenty major amino acids in nature, and each protein is made of a number of these amino acids. The combinations are infinite, providing enormous diversity to the protein realm.

A protein is a polar polymer: it has a left end and a right end, and polymerization actually occurs from left to right (from N-terminus to C-terminus, see below). Figure 2.1, "Peptidic Bond Formation by Condensation" shows that the chemical reaction at the basis of protein synthesis is a *condensation*. A protein is the result of the condensation of amino acids with each other in an orderly polar fashion. A protein has a left end, called *N-terminus; amino-terminal end* and a right end, called *C-terminus; carboxy-terminal end*. The left end is an amino group ($_2$HN–) corresponding to the non-reacted α-amino group of the very first amino acid of the protein sequence. Upon condensation of a new entering amino acid onto the first N-terminal one, the amino group of the entering amino acid reacts (nucleophilic attack) with the α-carboxyl group of the N-terminal amino acid. A water molecule is released, and the formation of an amide bond between the two amino acids yields a dipeptide. The right end of the dipeptide is a carboxyl group (–COOH) corresponding to the un-reacted α-carboxyl group of the last amino acid to have been "polymerized in".

The bond formed by condensation of two amino acids is an amide bond, also called—in protein chemistry—a *peptidic bond*. The elongation of the protein is a simple repetition of the condensation reaction shown in Figure 2.1, "Peptidic Bond Formation by Condensation", granted that the elongation *always* proceeds in the described direction (a new monomer arrives to the right end of the elongating polymer, and elongation is done from left to right).

$$_2HN-CH-COOH \quad _2HN-CH-COOH$$
$$\qquad\ \ |\qquad\qquad\qquad\ \ |$$
$$\qquad\ R_1\qquad\qquad\qquad R_2$$

$$_2HN-CH-CO \longrightarrow HN-CH-COOH$$
$$\qquad\ \ |\qquad\quad +\qquad\quad |$$
$$\qquad\ R_1\qquad H_2O\qquad\ \ R_2$$
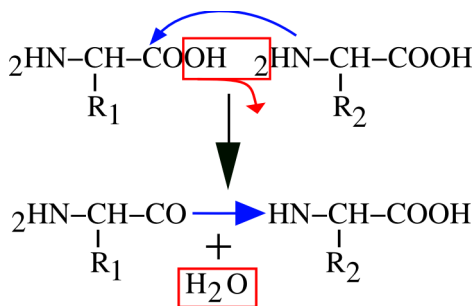
The left end monomer R_1 is condensed to the right end monomer R_2 to yield a peptidic bond. A water molecule is lost during the process.

FIGURE 2.1: PEPTIDIC BOND FORMATION BY CONDENSATION

Note

Now we should point at a protein chemistry-specific terminology issue: we have seen that a protein is a polymer made of a number of monomers, called *amino acids*. In protein chemistry, there is a subtlety: once an amino acid has been polymerized into a protein, it is no more called an amino acid, but is called a *residue* instead. We may say that a residue is an amino acid less a water molecule.

From what we have seen until now, we may define a protein this way: — *"A protein is a chain of* residues *linked together in an orderly polar fashion, with the residues being numbered starting from 1 and ending at n, from the first residue on the left end to the last one on the right end"*. This definition is still partly inexact, however. Indeed, from what is shown in FIGURE 2.2, "END CAPPING CHEMISTRY OF THE PROTEIN POLYMER", there is still a problem with the extremities of the residual chain: what about the amino group on the left end of a protein (the amino group sits right onto the first amino acid of the protein), and what about the carboxyl group of the right end of a protein (the carboxyl group sits right onto the last amino acid of the protein)? Because these groups lie at the extremities of the residual chain, they remained unreacted during the polymerization process. But because we are simulating a residual chain using residues and not amino-acids, we still need to put the protein polymer molecule in its "finished state": by *capping* the left end with a proton *cap* (so as to complete the amino group) and the right end with a hydroxyl cap (so as to complete the carboxyl group). The capping of the residual chain extremities ensures that the polymer is in its finished state, and that it cannot be elongated anymore. The proton is the *left cap* of the protein polymer and the hydroxyl is the *right cap* of the protein polymer.
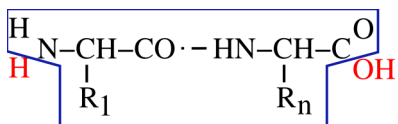
A protein is made of a chain of residues and of two caps. The left cap is the N-terminal proton and the right cap is the C-terminal hydroxyl. Altogether, the residual chain (enclosed here in the blue polygon) and both the H and OH red-colored caps do form a complete protein polymer in its finished state.

FIGURE 2.2: END CAPPING CHEMISTRY OF THE PROTEIN POLYMER

Now comes the question of unambiguously defining the structure of a protein. It is commonly accepted that the simple ordered sequence of each residue code in the protein, from left to right, constitutes an unambiguous description of the protein's primary structure (that is, its sequence). Of course, proteins have three-dimensional structures, but this is of no interest to a program like *massXpert*, which is aimed at calculating masses of polymers. To enunciate unambiguously the sequence of a protein, one would use a symbology like this:

- Using the 3-letter code of the amino acids:
  Ala Gly Trp Tyr Glu Gly Lys

- Using the 1-letter code of the amino acids:
  A G W Y E G K
  Alanine is thus the residue 1 and Lysine is the last residue (n = 7)

2.1.2   PROTEIN DISRUPTING CHEMISTRIES

The "polymer chain disrupting chemistry" was mentioned earlier as a complex subject that was of *enormous* importance to the mass spectrometrist. This is why that subject will be treated in a pretty thorough manner. First of all it should be noted that a chemical modification of a polymer does not necessarily involve the perturbation of the chain structure of the polymer. Here, however, we are concerned specifically with a number of chemical modifications that yield a polymer chain perturbation; *cleavages* and *fragmentations*:

Cleavages.   These are chemical processes by which a cleaving agent will act directly on the protein residual chain making it fall into at least two separated pieces (the peptides).

Fragmentations.   These are chemical processes by which the polymer structure is disrupted into separated pieces (the *product ions*, or *fragments*) mainly because of energy-dependent electron doublet rearrangements leading to bond breakage.

Protein Cleavage

Upon cleavage of a protein, the cleaving molecule reacts with it, and by doing so directly or indirectly *"dissolves"* an inter-residue bond. A protein cleavage always occurs in such a way as to generate a set of *true* finished polymerization state "proteins" (smaller in size than the parent polymer, evidently, which is why they are called *oligopeptides*, or *peptides*). Indeed, let us take the example shown in Figure 2.3, "Protein Cleavage by Water and Cyanogen Bromide", where a tripeptide (a very little protein, containing a methionyl residue at position 2) is submitted either to a water-mediated cleavage (hydrolysis, upper panel) or to a cyanogen bromide-mediated cleavage (lower panel). The two cases presented in this figure are similar in some respects and different in others:

- In the first case the molecule that is responsible for the cleavage is water, while in the second case it is cyanogen bromide;

- In both cases the bond that is cleaved is the inter-monomer bond (in protein chemistry this is a peptidic bond);

- In both cases the Oligomer 2 has the same structure;

- The structures of the Oligomer 1 species differ, when produced using water or cyanogen bromide as the cleaving molecule.

The difference between hydrolysis and cyanogen bromide cleavage is in the generation of the Oligomer 1 species: the cyanogen bromide cleavage has a side effect of generating a homoseryl residue at the C-terminus of Oligomer 1, while hydrolysis generates a genuine methionyl residue. This is because water reverses in a very symmetrical manner what polymerization did (hydrolysis is the converse of condensation), while cyanogen bromide did some chemical modification onto the generated Oligomer 1 species.
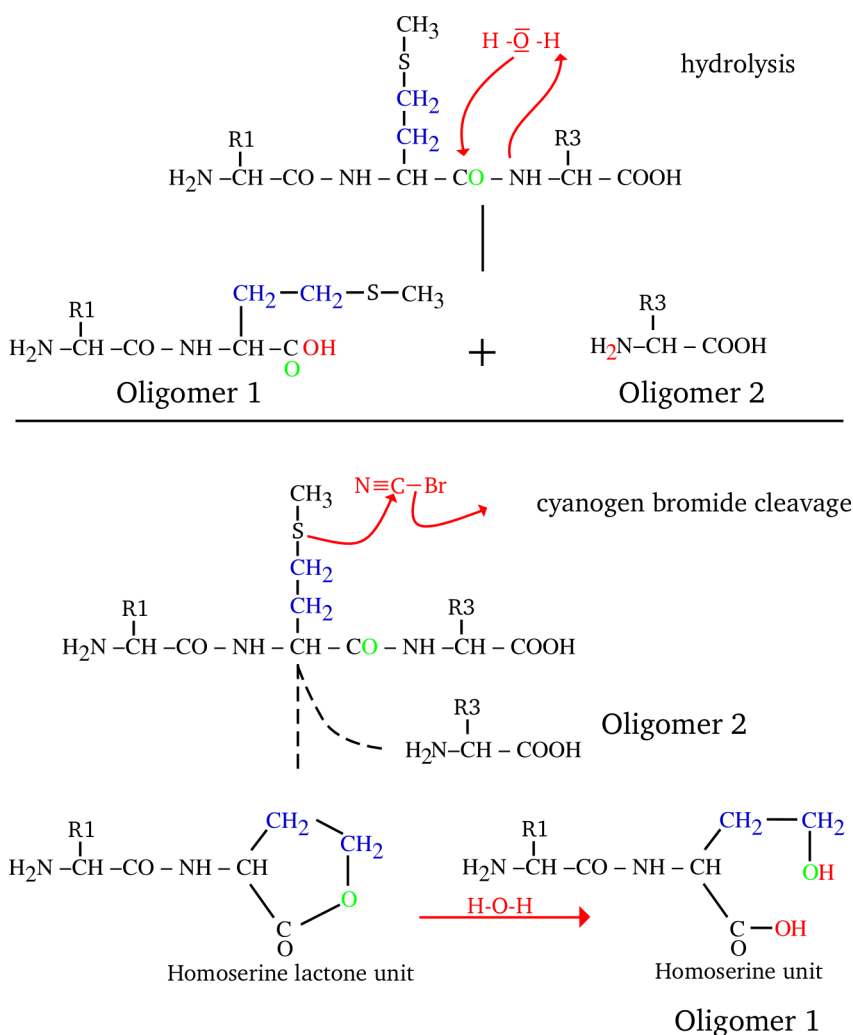
A tripeptide is cleaved at position 1 either by hydrolysis (top) or by cyanogen bromide (bottom). Cyanogen bromide cleaves specifically on the right of a methionine monomer. Upon cleavage, the methionyl monomer gets converted into homoserine by the cyanogen bromide reagent

FIGURE 2.3: PROTEIN CLEAVAGE BY WATER AND CYANOGEN BROMIDE

Nonetheless, the reader might have noted that—interestingly—all the four oligomers do effectively have their left cap (the proton, making the N-terminal amino group) and their right cap (the hydroxyl, making the C-terminal carboxyl group). This means that in both water- and cyanogen bromide-mediated cleavages, all the generated oligomers are indeed true polymers in the sense that: 1) they are a chain of residues (modified or not) and 2) they are correctly capped (*i.e.* they are polymers in their finished polymerization state). This is important because it is the basis on which we shall make the difference between a cleavage process and a fragmentation process. Thus, our definition of a peptide might be: *a peptide is a* protein *(of at least one residue) in its finished polymerization state that was generated upon cleavage of a longer protein*. Of course, when we use the term "*protein*", above, we mean "protein polymer", irrespective of its size.

When the protein cleavage reaction precisely reverses the reaction that was performed for the same protein's biosynthesis, there is no special difficulty. But when the cleavage reaction modifies the substrate, then this should be carefully taken into account when using *i2MassChroQ*. This is true for any chemical modification that happens onto a protein.

Well, all this sounds reasonable. But what about the "normal" case, when the cleavage is done using water? Nothing special: the mass of the oligomer is calculated by summing the mass of each monomer in the oligomer (since the monomers are not modified, this is easily done) and the masses corresponding to the left and right caps (these are defined in the polymer chemistry definition; in our present case it would be a proton on the left end, and a hydroxyl on the right end). In this way, the oligomer complies with its definition, which states that it is a faithful polymer made of monomers and that it is in its finished state.

Yes, but then how should one calculate the mass of the modified oligomer, like our Oligomer 1 in the case of the cyanogen bromide-mediated cleavage? Simple enough: in a first step it does exactly the same way as for the unmodified oligomer. Next, each oligomer is checked for presence or absence of a methionine residue on its right end. If a methionine is found, the mass corresponding to the "$-C_1H_2S_1+O_1$" chemical reaction is applied. And that's it.

2.1.2.2 Protein Fragmentation

In a fragmentation process, the bond that is broken does not necessarily yield smaller-sized "proteins" because fragmentation does not necessarily break the inter-residue bond the same way that the hydrolysis does. Indeed, fragmentations are oft-times high energy chemical processes that can affect peptidic bonds at different locations, not necessarily between the CO-NH bond of the peptidic bond. This is one of the reasons why fragmentations do differ from cleavages.

Another peculiarity of fragmentations, compared with cleavages, is the fact that there is no cleaving molecule starting the process, like water or cyanogen bromide, for example. Indeed, in the gas phase, the peptidic ions are "isolated": that is, very far one from each other. A fragmentation process is often initiated by an intra molecular electron doublet rearragement that propagates more or less in the polymer structure to eventually break it. Fragmentations are mainly a gas phase process, not some reaction that happens in solution as a result of putting in contact the polymer and some reagent. It is precisely because no cleaving molecule is involved in the fragmentation process that the obtained fragments are not necessarily capped like a normal polymer should be; and this is another really important difference between cleavage and fragmentation. The following examples should illustrate these concepts.

For the sake of completeness of this section, it must be noted that it is possible to have other "*chemical/physical* entities" intervene during the gas phase fragmentation process by enacting a chemical reaction, be these entities ions, electrons or photons. In bottom-up proteomics, the intervening molecules are gas molecules (nitrogen, most often, or helium) that act as physical entities imposing collisions to the peptidic ions with the effect that the ions acquire internal energy, eventually leading to dissociation (CID, for "collisionally-activated dissociation").

There is a pretty important number of different kinds of fragments that can be generated upon fragmentation of peptides. We are going to detail the most common ones.
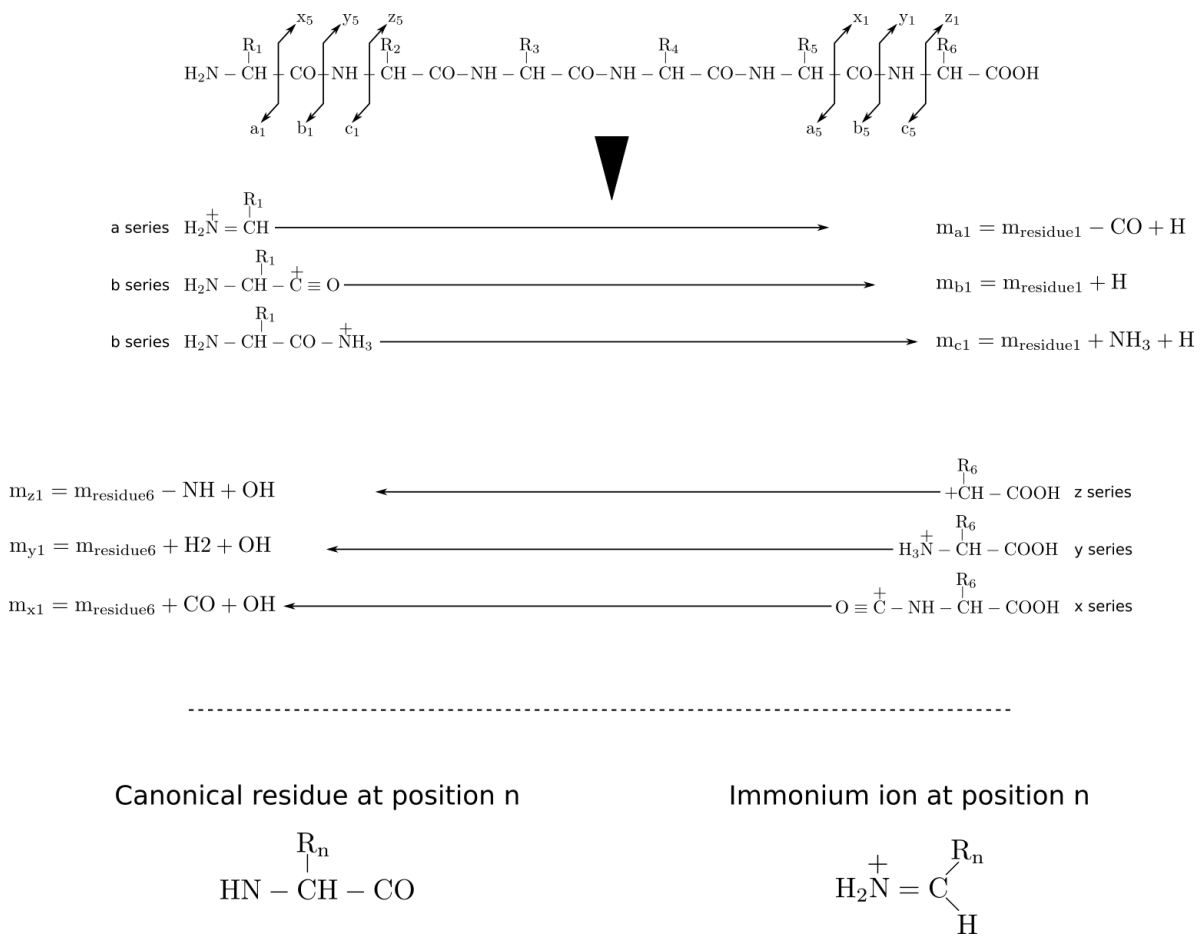
An hexapeptide is fragmented in the seven most widely encountered manners, such as to generate product ions of the a, b, c, x, y, z series and also immonium ions. The figure illustrates the position of the bond dissociation for each kind of fragment (exemplified using the case of the smallest fragment possible) and the mass calculation method is described for each fragment kind; consider that each fragment bears only *one positive* charge.

FIGURE 2.4: PROTEIN FRAGMENTATION PATTERNS MOST WIDELY ENCOUNTERED

As can be seen from FIGURE 2.4, "PROTEIN FRAGMENTATION PATTERNS MOST WIDELY ENCOUNTERED", the fragmentations do generate fragments of three categories: the ones that include the left end of the precursor polymer (a, b, c), the ones that include the right end of the precursor polymer (x, y, z), and finally the special case in which the fragment is an *internal fragment*, like the immonium ions. When looking at the fragmentations described in the figure, it becomes immediately clear why a fragmentation cannot be mistaken for a cleavage: the ionization of the fragment is not necessarily due to the captation of a proton by the fragment. Furthermore, we can also see that a fragmentation is not a cleavage because the fragment that is generated is *absolutely* not necessarily what we call a polymer, in the sense that the fragment might not be capped the same way as the precursor protein/peptide is (that is, the fragment is not in its finished polymerizaton state).

By looking at FIGURE 2.4, "PROTEIN FRAGMENTATION PATTERNS MOST WIDELY ENCOUNTERED", the reader should have noticed that the fragment naming scheme takes into consideration the fact that the fragment bears the N-terminal or C-terminal end of the precursor peptide (or none, also). Indeed, the numbering of fragments holding the N-terminal end of the precursor polymer sequence begins at the left end, and for fragments that hold the C-terminal end, at the right end. Thus the third fragment of series *a* (*a3*) would involve monomers [1→] and the third fragment of series *y* (*y3*) would involve monomers [6→] (see arrows in the figure).

2.2   GENERAL OVERVIEW OF BOTTOM-UP PROTEOMICS

Bottom-up proteomics is a field of endeavour where the ultimate goal is to identify the greatest number of proteins in a given sample. This goal might also, depending on the project at hand, be doubled with another goal: characterize at the finest level possible the nature and the position of post-translational/chemical modifications beared by the proteins.

To achieve the best results, proteomics has developed over the years a number of methods and techniques that, taken together, have allowed scientists to obtain impressive results of protein identification on pretty complex samples. These are listed below:

- *Mass spectrometers:* The development of mass spectrometers of ever-greater resolution power has allowed to attain at ever-lower false discovery rates over the years. In particular, the development of the Orbitrap analyzers, along with the huge improvements of the time-of-flight (TOF) mass analyzer technology, have strongly increased the identification results reliability by allowing the downstream data processing step to be more stringent in the protein identification task (see below);

- *Chromatography:* The development of highly resolutive chromatography resins along with the elaboration of hardware (columns, chromatography setups) that yields sensitivity improvements have had their share in the way proteomics has evolved over the years;

- *Bioinformatics:* The development and refinement of software that can cope with extremely large data sets (think metaproteomics) is one major field that enabled significant advances in proteomics. Also, refinement of algorithms related to the simulation of isotopic clusters and comparison with experimental data have had their part. Likewise so for algorithms that detect the charge of ions based on the analysis of the isotopic cluster peaks. Being able to single out without error the monoisotopic peak of an isotopic cluster (whater the ion charge or m/z ratio) is a big part of the successfully tackled challenges at the root of successful proteomics data processing.

In this section, we will review the bioinformatics-based mass spectrometric data processing, as it is the core subject of this user manual. In particular, we will provide an outline of how the major software packages on the market perform protein identification on the basis of mass spectrometric analyses of biological samples.

This section will outline in not-so-rough terms how bottom-up proteomics works, from the protein sample to the protein identification list. The workflow comprises two sequential processes:
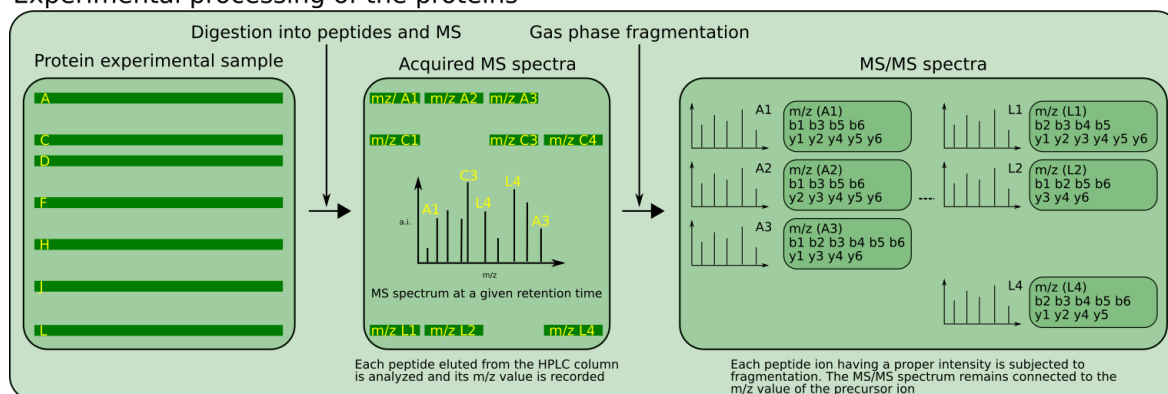
- *From the protein to the sequences of the peptides*: this initial part of the workflow is somehow doubled by having two parallel processes replicating it:

    - *In silico* process;

    - Experimental process.

    These two processes are described in Figure 2.5, "Theoretical and experimental parallel data-producing processes".

In silico processing of the proteins
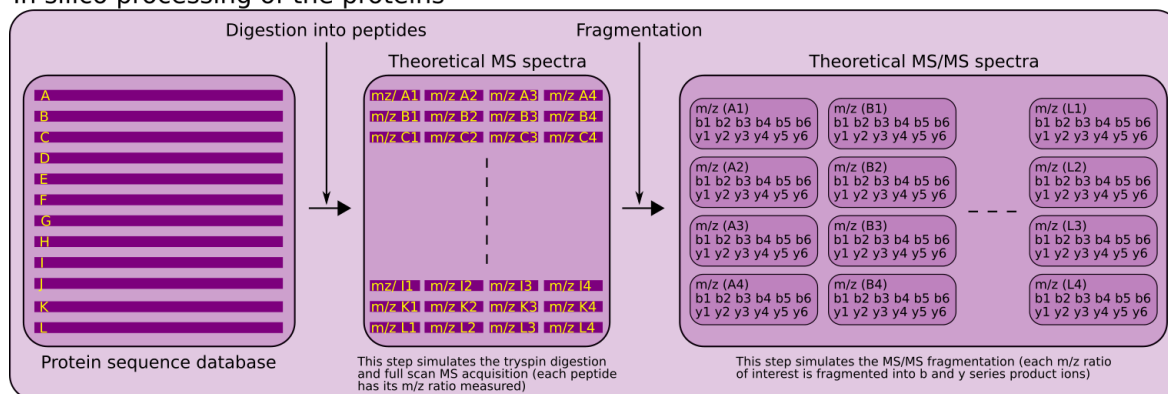
Experimental processing of the proteins

The digestion of the proteins, the analysis of the m/z of the peptides and the sequencing of the peptides are processes that exist both *in silico* and experimentally. This figure shows how the processes somehow mirror each other in the virtual and real contexts.

FIGURE 2.5: THEORETICAL AND EXPERIMENTAL PARALLEL DATA-PRODUCING PROCESSES

- *Database searching using experimental data:* this last part of the workflow is entirely based on bioinformatics software and involves the search for peptide *vs* mass spectrum matches and then a process called *protein inference* (see Section 2.2.5, "Matching Fragmentation Spectra with Theoretical Spectra").

2.2.1 The First Step: Digestion of the Sample's Proteins

The very first step in the bottom-up protemics workflow is to digest all the proteins in the initial biological sample with a site-specific endoprotease: typically tryspin.

The sample is subjected to proteolysis with all its proteins unresolved. This produces a highly complex mixture of peptides, each having a constant characteristic: each peptide has one predictable end (unless it is either the protein's N-terminal or the C-terminal peptide, as detailed below), either N-terminal or C-terminal:

- *Predictable N-terminus:* when the protease cuts at the N-terminal end of the target residue. For example, EndAspN cleaves left of Asp residues, thus producing peptides that always have Asp as their N-terminal residue. The only exception is when the peptide is the protein's N-terminal peptide and the first residue is not Asp);

- *Predictable C-terminus:* when the protease cuts at the C-terminal end of the target residue. For example, the most used enzyme, trypsin, cuts right of the basic residues Lys and Arg. The generated peptides thus necessarily end with one of these two residues. The only exception is when the peptide is the protein's C-terminal peptide and the last residue is not Lys nor Arg.

Tip

One interesting feature of trypsinolyis is that it generates peptides that—for their major part—will most probably be protonated twice: on their N-terminal end (the primary NH_2 amine group [1] and on the basic residual chain of the basic residue found at their C-terminal position (the ε-amine group for Lys and the guanidium group for Arg). Upon fragmentation of the peptide's precursor ion, both the left hand side fragment and the right hand side fragment will bear a proton and will thus be detected, thus potentially providing a better coverage of the peptide's sequence during the MS/MS experiment.

2.2.2 Chromatographic Separation of the Peptidic Mixture

One major analytic step in bottom-up proteomics is the separation of the peptides obtained by endoproteolysis of all the proteins in the sample. Indeed, analyzing all the peptides in one single injection without any prior chromatographic separation would yield catastrophic results, similar to having injected nothing in the mass spectrometer.

The typical method for resolving peptides is by separating them on a chromatographic column functionalized with a hydrophobic group (for peptides, that would be a C_{18} reversed phase column).

The chromatographic gradient that will elute the peptides progressively according to their increasing hydrophobicity will be developed over the 5–95 % of acetonitrile (a non-protic organic solvent).

Tip

Using acetonitrile as the non-protic organic solvent has the huge benefit of not injecting protons inside the mass spectrometer as the chromatographic gradient develops.

---

1 If not either converted to an amide group by acetylation or formylation or cyclised.

The eluate of the chromatographic column is directly injected into the mass spectrometer's source. The role of the mass spectrometer's source device is to ensure that the analytes are desolvated and ionized upon their entering in the core part of the mass spectrometer. Most often, that source is an electrospray source that is fed a liquid (typically, the eluate from the column). The source is designed to evaporate the solvent (analyte desolvation) and —having an electric potential applied to it— to help ionize the analytes (often the peptides are already ionized in solution, prior to desolvation). The electrically charged analytes in the gas phase are thus ions, the m/z (mass-to-charge) ratio of which can be measured by the mass spectrometer analyzer.

✋ WARNING

There are two main sources used in the mass-spectrometry-for-biology specialty: the matrix-assisted laser desorption ionization (MALDI) source and the electrospray ionization (ESI) source. One important difference between the two is that the MALDI process mostly produces mono-charged ions ($[M+H]^+$), while the ESI process mostly produces multi-charged ions ($[M+nH]^{n+}$). This has huge implications in the mass data analysis.

The source that is mainly used in bottom-up proteomics is the ESI source.

2.2.3   MASS SPECTROMETRIC ANALYSIS OF THE PEPTIDES

Upon elution off the chromatographic column, the peptides are desolvated, ionized and drawn into the mass spectrometer using an electrical field. Once they have entered the mass spectrometer they are analyzed in the mass analyzer of the instrument.

🔖 NOTE

There are a variety of mass analyzers commonly used in bottom-up proteomics. In fact, one single instrument might have as many as 4 or 5 mass analyzers. However, not all the analyzers in the instrument are responsible for the m/z measurement.

Sometimes, during the whole cycle of the analysis, two different mass analyzers are used at different steps of the cycle: one analyzer selects the ion for fragmentation and another analyzer measures the m/z value of the fragments.

In bottom-up proteomics, two different kinds of mass spectrometric data are required—ideally, for each peptide eluted from the column— in order to effectively identify the proteins in the initial sample:

- The mass-to-charge ratio value (m/z) of the peptide ion;

- The m/z values of the fragments (the product ions) of the peptidic precursor ion that has undergone an MS/MS gas phase fragmentation[2].

These two kinds of data are necessary because the protein identification process is based on searches in protein databases using the precursion ions' m/z value and the m/z values of that ion's fragments when it is fragmented. The way the protein databases are used as the substrate of these searches is described in the next section.

2.2.4 THE PROTEIN DATABASES AND THEIR USE

The previous section ended on the idea that the protein identification process, that is based on the analysis of all the peptides of a peptidic mixture resulting from the endoproteolysis of a sample containing many proteins, requires searches into protein databases.

A bottom-up proteomics experiment typically needs at least one protein database: a database listing all the known proteins of the organism from which the initial sample of proteins was prepared. That organism might be a bacterium, a Eucaryote, like a fungus, a protist, a plant, a mammalian... Optional databases might be used, like protein databases listing all known protein contaminants, for example.

The protein databases are files in the following FASTA format:

```
>GRMZM2G009506_P01 NP_001149383 serine/threonine-protein kinase receptor
MEEQHMAGPPYRYRLQHRRLMDIAPASASDDDSGHHGSNGMAIMVSILVVVIVCTLFYCV
YCWRWRKRNAVRRAQIERLRPMSSSDLPLMDLSSIHEATNSFSKENKLGEGGFGPVYRGV
MGGGAEIAVKRLSARSRQGAAEFRNEVELIAKLQHRNLVRLLGCCVERDEKMLVYEYLPN
RSLDSFLFDSRKSGQLDWKTRQSIVLGIARGMLYLHEDSCLKVIHRDLKASNVLLDNRMN
PKISDFGMAKIFEEEGNEPNTGPVVGTYGYMAPEYAMEGVFSVKSDVFSFGVLVLEILSG
QRNGSMYLQEHQHTLIQDAWKLWNEDRAAEFMDAALAGSYPRDEAWRCFHVGLLCVQESP
DLRPTMSSVVLMLISDQTAQQMPAPAQPPLFASSRLGRKASASDLSLAMKTETTKTQSVN
EVSISMMEPRFWADPGTSNGAATSHPATGACKKRGGQGGDRNVKDGLAARTPTHQPVARW
HHDRRIVD
```

This format is really simple, because it only contains three information pieces, grouped in as many stanzas as there are proteins in the database:

---

2 Most often, that fragmentation step is performed using collisionally-activated dissociation (CID). In this process, the peptidic precursor ion is first isolated in the gas phase on the basis of its m/z value and then is accelerated against a gas "fog" inside of the collision cell of the instrument. The ion hits gas molecules multiple times, acquires a lot of energy and finally breaks.

- The *unique* protein's accession id in the database (`GRMZM2G009506_P01`) that comes right after the '>' prompt that signals a new protein stanza;

- The protein description (`NP_001149383 serine/threonine-protein kinase receptor`) that provides some functional data bits for the protein at hand;

- The protein sequence (the rest of the stanza above).

The first (id) and second (description) information bits are used in various places in the *i2MassChroQ* program.

The protein databases are used by the protein identification software as the very first step in a bottom-up proteomics data analysis process: the proteins in the database are digested *in silico* in order to produce a list of peptides that retain a connection to the protein from which they were generated. For each one of all these peptides, the following data bits are computed (Figure 2.5, "Theoretical and experimental parallel data-producing processes", top panel):

- *sequence:* The peptide's sequence;

- *m/z value:* The peptide's m/z value, often computed for the mono-protonated ($[M+H]^+$) ion;

- *MS/MS spectrum:* The peptide's fragmentation spectrum is nothing but an array of m/z values corresponding to the set of calculated fragments (of the b and y ion series). The m/z values of the product ions are crucial for the database search algorithm;

The next step is the establishment of a relation between the experimental MS/MS data acquired by the instrument and the theoretical MS/MS spectra computed from the protein sequences in the database. This next step is described in detail in the next sections.

2.2.5   Matching Fragmentation Spectra with Theoretical Spectra

This section is about how the protein database searching software sets a relation between the experimental mass data and the theoretical mass data originating in the protein database. The elementary relation is between a given *experimental* MS/MS mass spectrum of a peptide's ion at a given m/z value and its *theoretical* counterpart from the database: when these two MS/MS spectra match at a sufficiently convincing level, then a *"peptide vs mass spectrum match"* was achieved (abbreviated name: PSM). The computing of a PSM is described in detail in Figure 2.6, "The steps leading to a scored peptide vs mass spectrum match (PSM)".

We have seen in Section 2.2.4, "The Protein Databases and Their Use", that two somehow similar processes are at the basis of the preparation of the data for the subsequent database searches. These processes were described in Figure 2.5, "Theoretical and experimental parallel data-producing processes".

On the one hand (top panel, violet), the protein database is processed to digest *in silico* every protein it contains into a list of peptides. For each peptide arising from the digestion of a protein, the following data elements are recorded:

- The peptide's m/z value is computed. The association between that m/z value, the peptide and its originating protein is maintained;

- The peptide is fragmented into a list of peptidic fragments (product ions' m/z values, that is, the MS/MS spectrum; typically b and y ions series). The connection with the earlier data elements above is also maintainted.

It is thus easy to determine the filiation between any given MS/MS theoretical mass spectrum, the precursor ion's m/z value, the peptidic sequence and, finally, the protein whence that peptide came.

On the other hand (bottom panel, green), the mass spectrometric data acquisition yields a huge set of the following pairs of data elements that are recorded over time:

- The m/z value of the peptidic precursor ion undergoing fragmentation (keeping a connection with the retention time at which it is recorded);

- The list of peptidic fragments (product ions' m/z values, that is, the MS/MS spectrum). The connection with the precursor ions' m/z value and with the retention time is maintained.
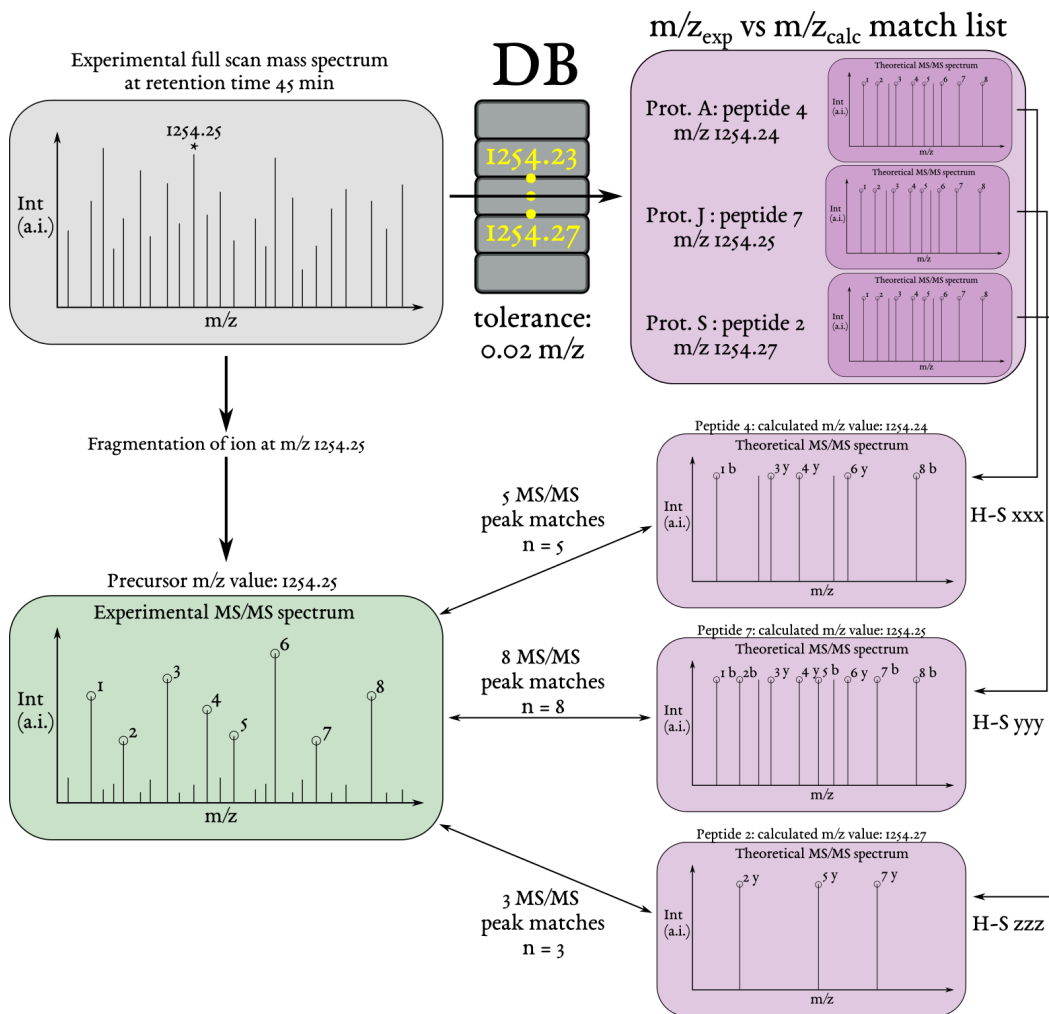
$$\text{X!Tandem HyperScore:} \left(\sum_{i=1}^{n} I_i * Pi \right) * N_b! * N_y!$$

n number of peak matches between the experimental and theoretical spectra

I_i intensity of the considered experimental peak

P_i ponderation of the considered experimental peak

$N_b!$ factorial of the number of matched peaks of the b ion series

$N_y!$ factorial of the number of matched peaks of the y ion series

H-S yyy >> H-S xxx >> H-S zzz

The process starts with a full scan mass spectrum from which the mass spectrometer selects one precursor ion at a definite m/z value. That ion is fragmented and thus generates a MS/MS spectrum. During the data exploration, the software extracts from the database all the peptides having the same m/z value as that of the fragmented ion (top right, violet background). Next, the experimental MS/MS spectrum is compared in turn to each one of the MS/MS spectra of the extracted peptide list. A HyperScore is computed at each comparison. Because *i2MassChroQ* uses *X!Tandem* as its preferred protein database search engine, the HyperScore calculation, as performed by *X!Tandem*, is described.

FIGURE 2.6: **THE STEPS LEADING TO A SCORED PEPTIDE *VS* MASS SPECTRUM MATCH (PSM)**

Once the acquisition of the experimental data is complete, the analysis of these data involves going through all the fragmentation data of the acquisition and performing these steps for *each* MS/MS spectrum (as evidenced in Figure 2.6, "The steps leading to a scored peptide vs mass spectrum match (PSM)":

- Get the precursor ion's m/z value;

- Compute the match m/z range. For example, if the software is configured with a m/z tolerance for the m/z matches set to 0.02 and the precursor ion's m/z value is 1254.25, then the match m/z range would be [1254.23–1254.27];

- Construct a list of all the peptides in the database that have their m/z value contained in the match m/z range;

- For *each* peptide in the list returned from the database, compare its theoretical MS/MS spectrum with the experimental one. Compute a HyperScore for comparison.

2.2.5.1   Computation of the PSM HyperScore

Of course, it is extremely rare that an experimental MS/MS spectrum matches fragment-by-fragment an identical theoretical spectrum. Most often, some theoretical product ions (MS/MS spectrum peaks) are missing from the experimental fragmentation spectrum. Also, there will almost certainly be dozens (if not hundreds) of peptides having a m/z value in the searched m/z range. Most certainly, the vast majority of these peptides are not of the right sequence (that is, do not have their MS/MS theoretical mass spectrum matching the experimental one). To make without any human scrutiny of the matches, it is necessary to compute a score that somehow assesses the extent to which both the experimental and theoretical MS/MS spectra match. That score, in *X!Tandem*, is called *HyperScore* and is described at the bottom of the figure.

The HyperScore computation process is relatively straightforward. First off, it is necessary to stress the fact that a HyperScore is computed each time an *experimental* MS/MS spectrum is compared to a theoretical (*calculated*) MS/MS spectrum (see m/z_{exp} vs m/z_{calc} *match list* in Figure 2.6, "The steps leading to a scored peptide vs mass spectrum match (PSM)").

In the example, three peptides from the database have their m/z value matching the searched m/z range (the m/z value of the precursor ion with accounting for the tolerance). So, the program checks the similarity between the experimental MS/MS spectrum and each one of the three theoretical ones. Each similarity test is associated to a HyperScore value.

The HyperScore is computed by summing—for each tested fragment peak *in the theoretical MS/MS spectrum*—the product of two variables described below. Once that sum is computed, it is compounded by two factorial numbers also described below:

- I_i: the intensity of the matching mass peak in the experimental MS/MS spectrum (if found);

- P_i: the ponderation factor of the matching mass peak in the experimental MS/MS spectrum. That variable can take a number of values, depending on the presence or not of this fragment peak in the experimental MS/MS spectrum (if not found, then P_i is naught and the peak is disregarded entirely). There are other values greater than naught, accounting for the physico-chemical properties of the peptidic bond that was cleaved to obtain that fragment (presence of proline will lower the P value, for example).
  Intuitively, the HyperScore will end up larger if there are a lot of fragment peaks in the theoretical MS/MS spectrum that are matched with experimental ones (each P_i value compounded by the I_i value is being summed into the HyperScore final value).

- $N_b!$: the sum computed above is then compounded by the factorial of the number of ions of the b ion series that are found in the experimental MS/MS spectrum;

- $N_y!$: the product computed at the previous step is then compounded by factorial of the number of ions of the y ion series that are found in the experimental MS/MS spectrum.
  This last compounding operation terminates the computation of the HyperScore value.

It is apparent now that the HyperScore value will tend to be greater if there are numerous fragment peaks in the theoretical MS/MS spectrum that are matched by fragment peaks in the experimental MS/MS spectrum. Also, the score value is incremented if the intensity of the matching peaks is greater and if the number of matching peaks of the two b and y ions series is greater.

This, however, cannot be all of it, because the HyperScore does not really answers the question: *"what are—if any—, of all the PSMs found for a given experimental MS/MS spectrum, the one (or ones) that we can faithfully tell as true match(es)?"*. To answer that question, some more computational steps need to be carried over, that should lead to a numerical value that is truly indicative of the confidence we may have that a given PSM is a *real* match. In *X!Tandem*, that numerical value is called *expectation value* (abbreviation: *E-value*). We describe the whole process of its computation below.

2.2.5.2 COMPUTATION OF THE PEPTIDE EXPECTATION VALUE (E-VALUE)

First of all, it needs stating that we describe the *peptide E-value*, not the protein E-value. A peptide E-value is obtained for a single experimental MS/MS spectrum. It is computed by looking into the HyperScore values obtained for all the MS/MS spectra comparisons described at the previous section. The HyperScore values (for example, the three values denoted *H-S xxx*, *H-S yyy* and *H-S zzz* in FIGURE 2.6, "THE STEPS LEADING TO A SCORED PEPTIDE VS MASS SPECTRUM MATCH (PSM)") are used to perform the E-value computation. In the following text, we'll assume that there are many more PSMs than these three, for a given experimental MS/MS spectrum (which is actually the reality, with hundreds of peptides in the database that match a given searched m/z range). As illustrated in FIGURE 2.7, "COMPUTATION OF A PEPTIDIC EXPECTATION VALUE (E-VALUE)", a histogram is crafted

plotting the count of MS/MS spectral pair comparisons (let us call them "wannabe PSMs") against a number of HyperScore bins. This histogram is a good representation of the distribution of the HyperScore values among the various peptides in the m/z value-matching list (see previous section). In this example, the very best HyperScore value is 82 and the number of PSMs having that score is obviously very low! Instead, the distribution clearly shows that there are a vast majority of wannabe PSMs that have very low HyperScore values and that will not ultimately be considered as real PSMs.

In order to be able to use the distribution pattern further, the second half of the distribution's main peak is replotted by computing the natural logarithm of the count of MS/MS spectral pair comparisons, still against the HyperScore value bins. The new plot is easily fitted into a line, of which the equation is computed.

The best HyperScore value (82, in the example) is then used in the line equation to compute the corresponding ordinate (the natural logarithm of the PSMs count having that HyperScore). That value (-8.4, in the example) corresponds to the natural logarithm of the expectation value (E-value). By using the exponential function, the E-value is thus computed to be 0.00022, which a pretty low number. Since the E-value somehow gives an idea that a given PSM was obtained by chance, the very small obtained result shows that the match almost certainly was a faithful one.

NOTE

The expectation value is defined as the probability that the peptide sequence would match an experimental tandem mass spectrum by chance, if the trial is repeated many times. For example, if the E-value is found to be 1, then that means that the match can occur by chance or not with an equal probability. Instead, if the E-value is found to be 0.01, then that means that there is one event over 100 trials that the match has occurred by chance.

The smaller the E-value, the more confidence one has that the match is correct and that the PSM is a faithful one.

Precursor m/z value: 1254.25

Experimental MS/MS spectrum

Int (a.i.)

m/z

List of PSMs

List of HyperScore values

H-S 1
H-S 2
H-S 3
⋮
H-S 798
H-S 799
H-S 800

Distribution of the number of PSMs against the HyperScore value

Best HyperScore: 82

Very small number of PSMs having a high HyperScore

Number of PSMs

HyperScore value

Conversion to the natural logarithm

$f(x) = -0.2 \ast x + 8$

$f(82) = -8.4$

$-8.4 = \ln \text{(E-value)}$

$\text{E-value} = e^{-8.4}$

$\text{E-value} = 0.00022$

ln(Number of PSMs)

HyperScore value

For each experimental MS/MS spectrum, gather all the peptides in the database that have a m/z value matching the precursor ion's m/z value. For each peptide sequence, compute the HyperScore. With all the HyperScore values, go on with the calculation of the expectation value for the peptide set. The peptidic E-value should be the smallest possible, as it is an indication of the possibility that the match between the experimental MS/MS spectrum and the theoretical mass spectrum occurred by chance.

FIGURE 2.7: COMPUTATION OF A PEPTIDIC EXPECTATION VALUE (E-VALUE)

The user configures the software to only consider PSMs if their peptidic E-value is below a given threshold. Typically, that threshold is given a value of 0.05 (Figure 3.6, "Configuration of the loading of the identification results").

When a reliable match between an experimental MS/MS spectrum and a theoretical MS/MS spectrum is found (that is, a true PSM), the software reports the following set of data elements:

- *m/z:* the m/z value of the precursor peptidic ion that underwent fragmentation;

- *sequence:* the sequence of the peptide that was matched in the present PSM;

- *protein name:* the protein accession number that produced the matched peptide upon enzymatic digestion of the sample;

- *E-value:* the peptide expectation value, as described above.

2.2.5.3   Computation of the Protein Expectation Value (E-value)

The last step in the computation of values that help the software and the user determine if identifications are faithful (for peptides and for proteins) is the computation of the protein expected value. This value is very easily computed: it is the product of the E-values of all the peptides that participated in the identification of the protein.

By necessity, then, the protein E-value will be less than the threshold peptide E-value (since that last value is below 1). By default, the protein E-value is set to 0.01 (Figure 3.6, "Configuration of the loading of the identification results").

2.2.5.4   Protein Inference: from PSMs to Protein Identities

One remaining critical question is: *"— How is the list of protein identifications returned by the database searching software verified and modified?"* Indeed, there are a number of situation where the proteomics data user may want to tweak the identification results. But also, the protein identification list returned by the database software may not be as perfect as one would expect. Bioinformaticians working in proteomics have come up with a number of algorithms to better the reliability of the identification results returned by database searching software.

In *i2MassChroQ* we use an algorithm that is impinged on the concept of *parcimony*. That algorithm is detailed in an article describing *X!TandemPipeline-Java* that was published in *The Journal of Proteome Research* in 2017 by Olivier Langella and Colleagues. The general concepts are presented here for the sake of completeness of this user manual.
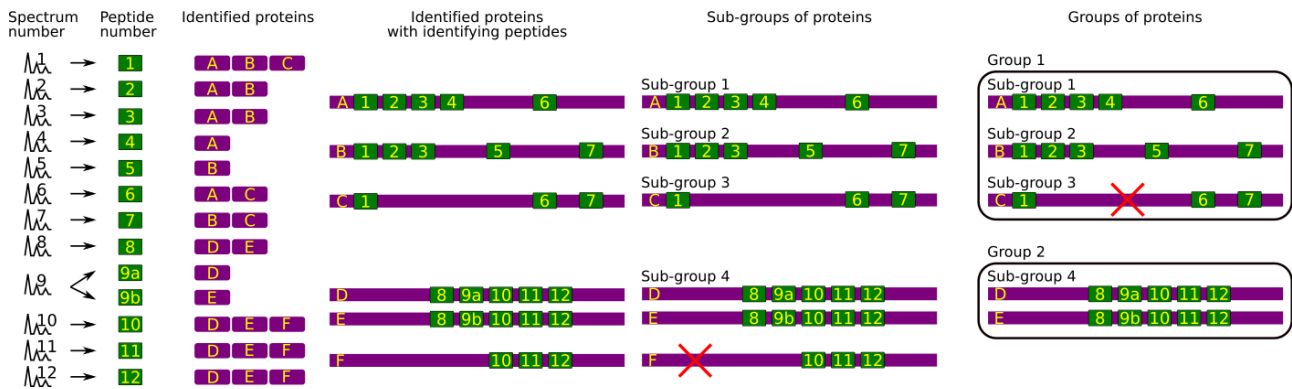
The process of establishing a consolidated protein identity list from the results reported by the database searching software is illustrated (see text).

FIGURE 2.8: PROTEIN INFERENCE: CONSTRUCTING A CONSOLIDATED PROTEIN IDENTIFICATIONS LIST

The protein inference process, depicted in FIGURE 2.8, "PROTEIN INFERENCE: CONSTRUCTING A CONSOLIDATED PROTEIN IDENTIFICATIONS LIST", is a multi-step one. The starting point is the huge list of PSMs that are reported by the database searching software. These PSMs are displayed in the figure as the two columns on the left hand side: one *experimental* MS/MS spectrum (*Spectrum number*) has provided a convincing PSM and thus allowed the identification of a peptide (MS/MS 1 → Pep 1, *Peptide number*). Of course, a given peptide (Pep 1) might have allowed the identification of multiple proteins (for example, homologous proteins that share the same peptidic sequence). Thus, Pep 1 is found in proteins A, B and C (column *Identified proteins*). The structure of the identified proteins can thus be partially reconstructed, and that is shown in column *Identified proteins with identifying peptides*. All the other PSMs are listed below that first one.

The general concept of the algorithm is that, by going through all the PSM data it is possible to check if some form of degraded redundancy allows pruning off some proteins from the list. This pruning off of some proteins is meant to increase the confidence that the identifications are reliable. That might be at the cost of having a smaller number of identified proteins, but with an improved false discovery rate (that is, a reduced FDR). As described below, the pruning off of proteins from the protien identifications list occurs at two different steps in the inference process.

The first step is the creation of sub-groups of the identified proteins. In this step, all the proteins that could be identified thanks to the exactly same set of peptides are gathered into a sub-group. In the example, the sub-group that contains more than one protein happens to be sub-group 4. Note how protein F in this sub-group is identified by a set of three peptides. This is two peptides less than the number of peptides that identified the other two proteins (D and E) in the sub-group. The principle of parcimony allows thus to remove Protein F as that protein is not justified *per se*, that is, it is unnecessary to explain the presence of the three peptides.

The second step is the creation of groups that gather all the sub-groups that share at least one peptide. Thus, group 1 contains sub-groups 1, 2 and 3, while group 2 contains the sub-group 4. According to exactly the same philosophy as for the previous step, the sub-groups that contain proteins identified only by peptides also shared by proteins present in other sub-groups are pruned off.

The whole process described here is dubbed "*protein grouping*" in the *i2MassChroQ* language. The ouptput of this protein grouping process is displayed in the protein identification window, to be described below.

2.2.6 Phospho-Proteomics

In this section, the typical procedures involved in phospho-proteomics projects are described, from the sample handling to the post-translational modification data exploration.

2.2.6.1 Handling Phospho-Proteomics Samples

i2MassChroQ is able to cope with phospho-peptides. The mass spectrometric data are acquired exactly as usual with the mass spectrometer, but the sample preparation goes along theses steps:

- Separate digestion of the samples (when there are more than one);

- Labeling of the peptides, each sample gets a different label;

- Pool of the whole set of peptides into a single mixture;

- Separation of the peptides on a strong cation exchange (SCX) resin, collection of the fractions;

- Phospho-peptide enrichment using IMAC[3] for each SCX fraction. The SCX fraction is loaded onto the IMAC resin and, following a wash step, the phospho-peptides are eluted (pH-based elution). There is thus a one-to-one relation between a SCX fraction and an IMAC-based purification fraction.

- Mass spectrometric analysis of each IMAC-based phospho-peptide-enriched fraction.

X!Tandem needs to be configured in such a manner that it can generate all the theoretical peptides (and fragments) that might bear the phosphoryl group. This process is described in the section below.

2.2.6.2 Protein Identification in Phospho-Proteomics Projects

An analogous algorithm as the one used for protein inference is at play when *i2MassChroQ* is handling phospho-proteomics data. That algorithm is described below and in Figure 2.9, "Phospho-site inference: constructing a consolidated phospho-site list".

---

3 Immobilized-metal affinity chromatography.
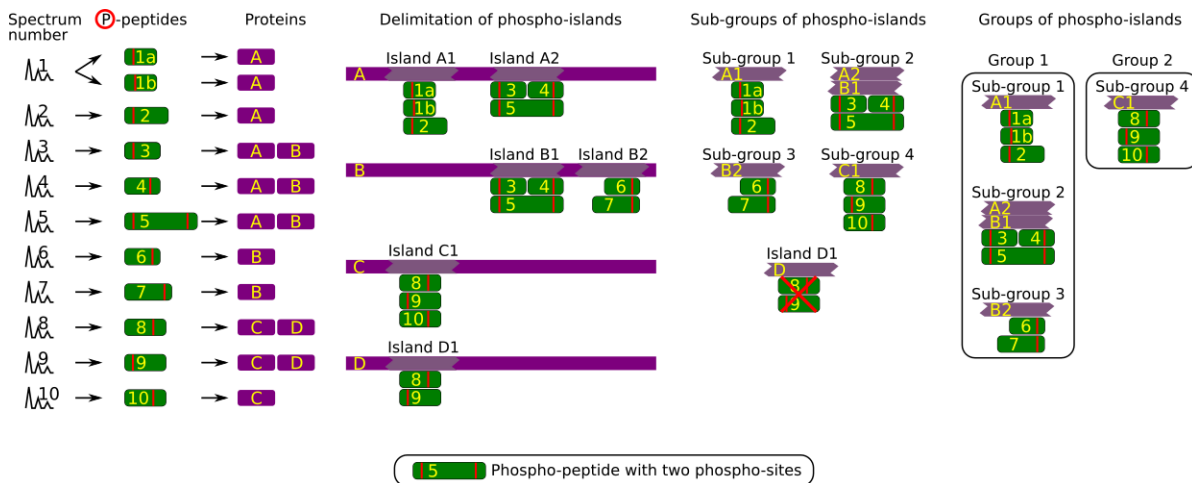
The process of establishing a consolidated phospho-site list from the results reported by the database searching software is illustrated (see text).

FIGURE 2.9: **PHOSPHO-SITE INFERENCE: CONSTRUCTING A CONSOLIDATED PHOSPHO-SITE LIST**

The phospho-island inference process, depicted in Figure 2.9, "Phospho-site inference: constructing a consolidated phospho-site list", is a multi-step one, most similarly to what was described in Section 2.2.5.4, "Protein Inference: from PSMs to Protein Identities". The starting point is the list of peptides that were identified and determined to bear one or more phospho-sites (thus called phospho-peptides; see the red vertical bar in the figure). Two difficulties here are, on the one hand, the fact that phospho-sites may be shared by more than one peptide and, on the other hand, the fact that more than one phospho-site might be determined on the *same* peptide. These are the reasons that the concept of *phospho-island* was elaborated: it is a protein region that bears at least one phospho-site, in turn beared by one or several overlapping phospho-peptides. It is important to note that the position and number of phospho-sites are not necessarily the same in all of the overlapping phospho-peptides.

In this inference process, the analogy with the previously described one is the following:

- Peptides are replaced by phospho-peptides;

- Proteins are replaced by phospho-islands.

In the first step, the phospho-islands are delimited on the phosphorylated proteins. In the second step, sub-groups of phospho-islands are created using all the phospho-islands identified in different proteins and that share exactly the same set of phospho-peptides. At this step, any remaining phospho-island defined by a subset of phospho-peptides only partially defining a sub-group is disregarded. In the example, phospho-island D1 is defined by two phospho-petides, 8 and 9, that also are part of a sub-group defined by these two peptides but also by phospho-peptide 10. Phospho-island D1 is thus disregarded.

In the third step, all the sub-groups that contain phospho-islands beared by the same protein are gathered in a group.

3 The main program window

Proteomics data explorations, with *i2MassChroQ*, entail, for a large part, the following steps:

- Configuration of the *X!Tandem* external software that runs the database searches (producing peptide vs mass spectrum matches—PSMs—, leading to the peptide identifications and ultimately to protein identifications);

- Configuration of the protein database files (both the organism-specific protein databases and optional contaminant-containing databases);

- Loading of the mass spectrometry data acquisition files (the mzML format is recommended);

- Running *X!Tandem* from inside of *i2MassChroQ*;

- Loading of the identification results produced during the previous step;

  ## Note

  i2MassChroQ can also handle peptide vs spectrum matches data (peptide identification data) from other software with the following formats:

  - mzIdentML;

  - pepXML;

  - Mascot DAT files

- Relentless scrutiny of the peptide identification results. Optional modification of the results;

- Protein inference, that is, protein identification on the basis of the peptide identifications. *i2MassChroQ* implements a protein grouping algorithm, as described in Figure 2.8, "Protein inference: constructing a consolidated protein identifications list", that leads to consolidated protein identifications. The program has an interface geared towards the tweaking of the protein grouping process so as to let the user in full control of the stringency with which the protein identifications list is ultimately generated.

In this chapter, *i2MassChroQ*'s main window's user interface is described in detail, in particular in the way it is a starting point for the main tasks briefly mentioned above.

3.1   STARTING A NEW *i2MassChroQ* WORKING SESSION

To start a session, run *i2MassChroQ* and the main program windows shows up as described in FIGURE 3.1, "MAIN PROGRAM WINDOW".
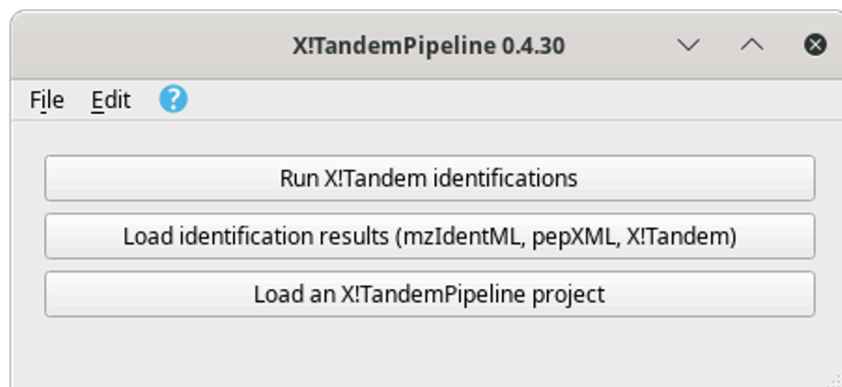
The main program window contains three buttons described in detail in the text.

FIGURE 3.1: MAIN PROGRAM WINDOW

The main program window contains three buttons that start the following main tasks:

- *Run X!Tandem identifications*. See SECTION 3.2, "RUNNING X!TANDEM IDENTIFICATIONS".

- *Load identification results (mzIdentML, pepXML, Mascot, X!Tandem)*. See SECTION 3.4, "LOADING THE PROTEIN IDENTIFICATION RESULTS".

- *Load an i2MassChroQ project*. See SECTION 3.5, "LOADING i2MASSCHROQ PROJECTS".

3.2   RUNNING *X!Tandem* IDENTIFICATIONS

To run *X!Tandem*-based identifications, click onto the *Run X!Tandem identifications* button. This triggers the opening of the window pictured in FIGURE 3.2, "X!TANDEM-BASED IDENTIFICATION CONFIGURATION".

The configuration of a *X!Tandem* run is performed in this configuration window (see text for details).

FIGURE 3.2: *X!TANDEM*-BASED IDENTIFICATION CONFIGURATION

The configuration of an *X!Tandem* run entails defining the following:

- *Configure the X!Tandem execution*: This setting allows one to specify the path to the *X!Tandem* software program. The version of the program, if found, is displayed below (in this case, *Alanine 2017.2.1.4*). This feature is useful when the user wants to test multiple versions of the *X!Tandem* software.

- *Run X!Tandem through HTCondor*: Only check the box if running *X!Tandem* over the network on a server supporting *HTCondor*[1].

- *Choose presets*: This setting defines the parameters that *X!Tandem* must use. Either load already known presets from the drop-down list widget or edit them (or create a new set) by clicking onto the *Edit* button. Note that to load an existing presets file, it might be necessary to point *i2MassChroQ* to the directory that contains the presets file. Use the folder icon for this, as visible in Figure 3.3, "X!Tandem presets configuration window (Spectrum tab)".

- *Choose database files*: Add protein database files in the FASTA format. There must be at least one protein database that contains all the known proteins for the organism of interest (there might be as many such database files as necessary) and optionally protein databases containing known contaminant proteins (there might be as many such database files as necessary). Click onto the *Clear list* button to clear the database files list and start anew if an error occurred (it is not possible to remove files one at a time).

- *Choose MS data files to process*: Add the mass spectrometry data files (mzML or mzXML format) to be processed by the *X!Tandem* software. As many files as necessary might be added in the list.

> ## 💡 Tip
>
> When using Bruker timsTOF data, click onto the *Add Bruker timsTOF folders* button to select folders containing this kind of data. Bruker timsTOF data come as two files that must sit in the same directory.

- *Output directory*: This setting specifies the directory into which new files output by the *X!Tandem* process need to be created. *X!Tandem* produces identification results in files in an XML format that *i2MassChroQ* reads during a later step.

- *Number of threads*: This setting defines the maximum number of execution threads that *X!Tandem* might be using during its run.

> ## 💡 Tip
>
> Although *i2MassChroQ* sets that number of execution threads to 1, it is beneficial to set that number to the highest value possible.

---

1 See HTTPS://RESEARCH.CS.WISC.EDU/HTCONDOR/ .

3.3 Setting the *X!Tandem* Run Presets

The *Edit* button of the *Choose presets* group box described above triggers the opening of a dialog window where the user might configure in the most detailed way the *X!Tandem* parameters. That dialog window is pictured in Figure 3.3, "X!Tandem presets configuration window (Spectrum tab)". Only the *Spectrum* tab is shown, but the interface is similar for all the other ones.
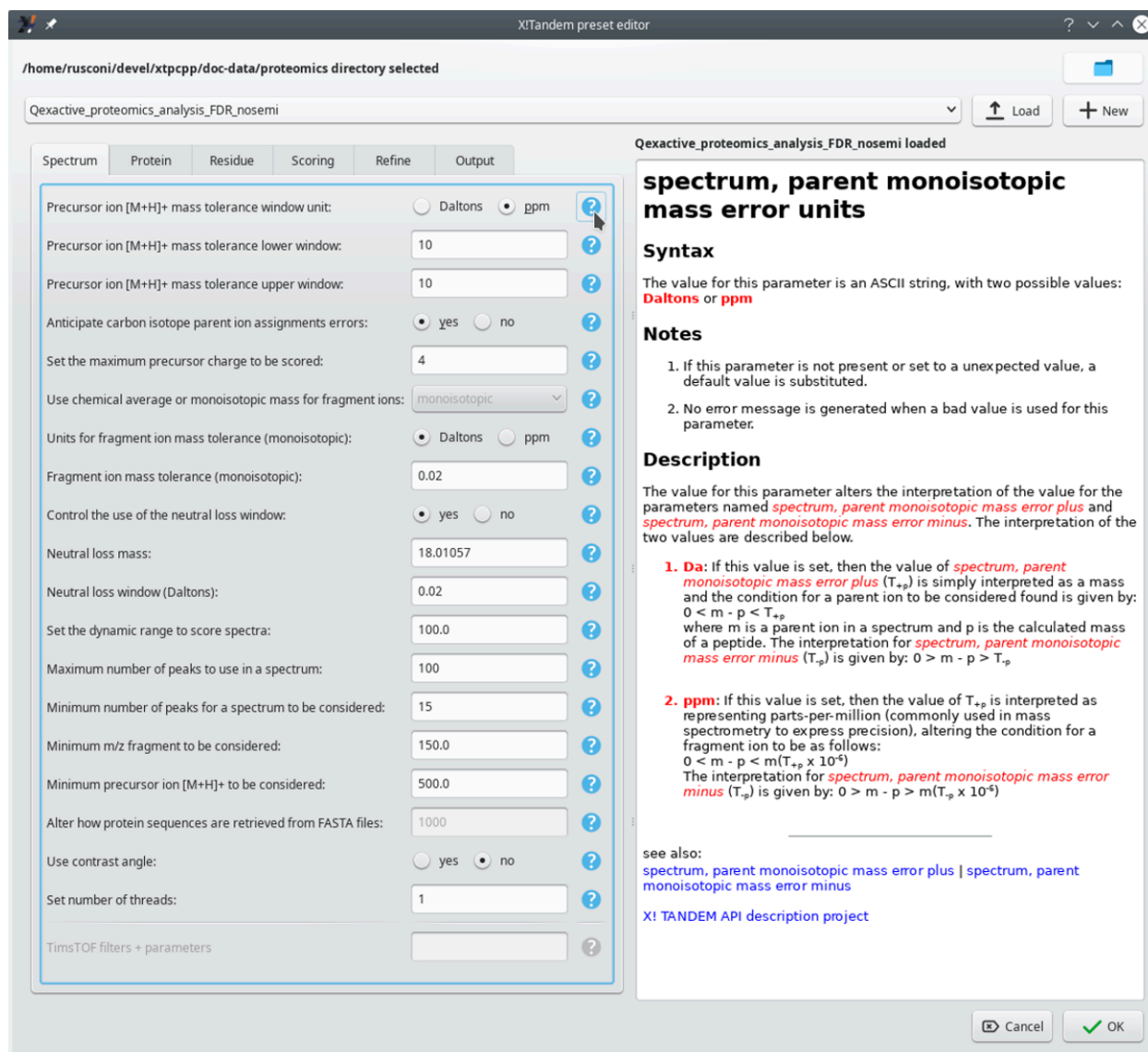
The configuration of the *X!Tandem* presets is performed in this configuration window. This window has its *Spectrum* tab selected. Each parameter is associated to a manual page[2] that can be displayed by clicking on the interrogation mark button next to it. It is possible to load existing presets from file or to create brand new ones.

FIGURE 3.3: *X!TANDEM* PRESETS CONFIGURATION WINDOW (*SPECTRUM* TAB)

---

2 The help texts are extracted automatically from the *X!Tandem* software documentation.

3.3.1 Loading existing presets configurations from file

It is possible to load existing *X!Tandem* presets (which is useful in particular if the samples most often come from the same instrument using the same configuration). To this end, first point *i2MassChroQ* to the right directory that contains the presets file of interest (click onto the folder icon at the top right corner of the window shown in Figure 3.3, "X!Tandem presets configuration window (Spectrum tab)"). The presets files in the chosen directory are automatically detected and listed in the drop-down list widget. At this point, select from that list the file of interest and click onto the *Load* button.

✋ **Warning**

It is compulsory to click onto the *Load* button to confirm loading of the presets file contents, because these are not updated upon choosing the file name from the drop-down list only.

3.3.2 Creating new presets configurations

It is possible to create a new presets file by clicking onto the *New* button. This opens an input dialog window for the user to provide a new file name (the edit widget is preset with the currently loaded file's name suffixed with *_copy*).

💡 **Tip**

One interesting feature of the new presets file creation process is that, if presets are already loaded, *i2MassChroQ* copies the currently displayed settings to the new file. From there, it is possible to create a variant *X!Tandem* presets file, which eases the exploration of the right *X!Tandem* parameters for a given sample data set.

3.3.3 Actual *X!Tandem* presets configuration

The dialog window pictured in Figure 3.3, "X!Tandem presets configuration window (Spectrum tab)" contains a number of tabs where various aspects of the *X!Tandem* run settings are handled. Each parameter's documentation can be seen on the pane on the right hand side of the window by clicking onto the question mark button next to it. These manual pages are authoritative because they are taken from the *X!Tandem* software package with no transformation whatsoever.

Once the configuration has been performed, click onto the *OK* button. If the parameters were modified, *i2MassChroQ* asks if they should be stored in the file.

3.3.4 Running a properly configured *X!Tandem* process

Once the *X!Tandem* settings configuration dialog window has been closed, it is possible to run *X!Tandem* from inside *i2MassChroQ* by clicking onto the *Run* button at the bottom of the window pictured in Figure 3.2, "X!Tandem-based identification configuration".

While the computation is carried over, the program shows the feedback dialog window pictured in Figure 3.4, "X!Tandem run feedback to the user".
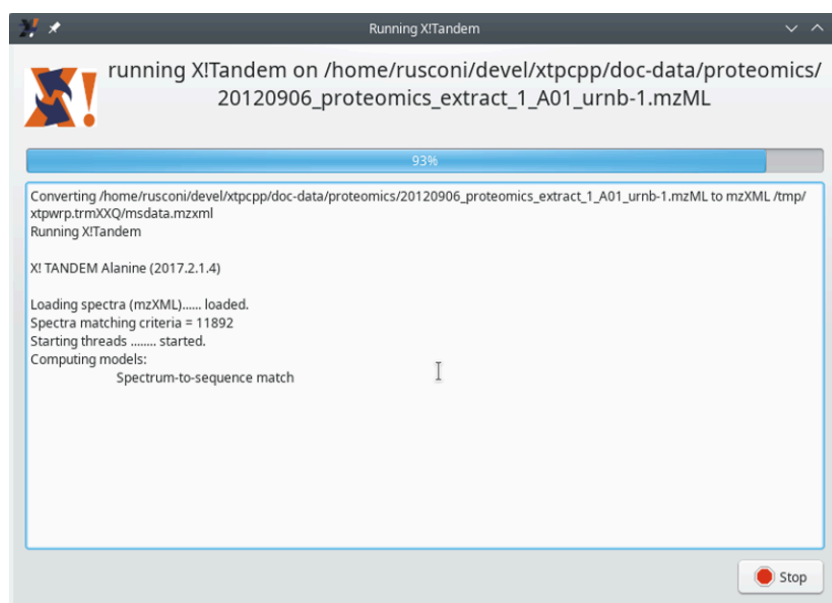
The text in this feedback dialog window is getting incrementally printed all along the computation.

FIGURE 3.4: *X!TANDEM* RUN FEEDBACK TO THE USER

Once the computation is finished, the feedback dialog window closes and the user is returned to the main program window (Figure 3.1, "Main program window") albeit with a message shown in Figure 3.5, "X!Tandem run finished message to the user".
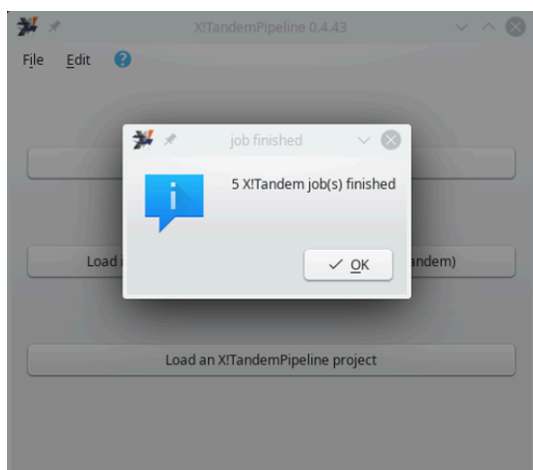
The *X!Tandem* run is now finished. Click the *OK* button to access the main program window.

FIGURE 3.5: *X!TANDEM* RUN FINISHED MESSAGE TO THE USER

From the main program window, it is possible to open the *X!Tandem* results file(s) located in the output directory configured above. There are as many output files (XML-based format, and xml extension) as there were mass spectrometry data files to process. The loading of the results files is carried over by first clicking the button labelled *Load identification results (mzIdentML, pepXML, X!Tandem)*. The process is described in Section 3.4, "Loading the Protein Identification Results".

3.4   Loading the Protein Identification Results

The loading of identification results comes with a minimal set of configuration required to instruct *i2MassChroQ* on the way to handle contaminant proteins, for example. This process is pictured in Figure 3.6, "Configuration of the loading of the identification results" and is described in the following section.

Loading identification results comes with some configuration that is described in the text.

FIGURE 3.6: CONFIGURATION OF THE LOADING OF THE IDENTIFICATION RESULTS

3.4.1 IDENTIFICATION DATA LOADING CONFIGURATION

- *Results handling mode*: there are two possibilities:

  - *Combine*: in this mode, all the identification results coming from different identification results files are merged into a single set. That single set is the basis for the protein inference step and the identified proteins are listed into a single protein list window.
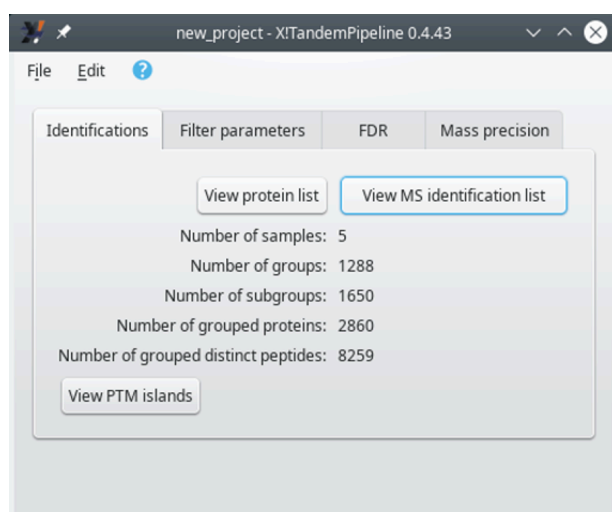
When loading multiple identification results files in *Individual* mode, the selection of any given identification results file is performed by selecting its name from the drop-down list widget *and* by clicking onto the *View protein list* button. Note that some metadata about the identifications are updated beneath the drop-down list widget.

FIGURE 3.7: SELECTING A PARTICULAR IDENTIFICATION RESULTS FILE'S DATA SET

- *Individual*: in this mode, the identification results coming for various files are kept separated. Thus, the identification results coming from each file are used for a separate protein inference step. The identified proteins list is thus displayed for *each single file* in turn. The selection of the file for which the protein list needs to be displayed is done via the main program window that changes its appearance:
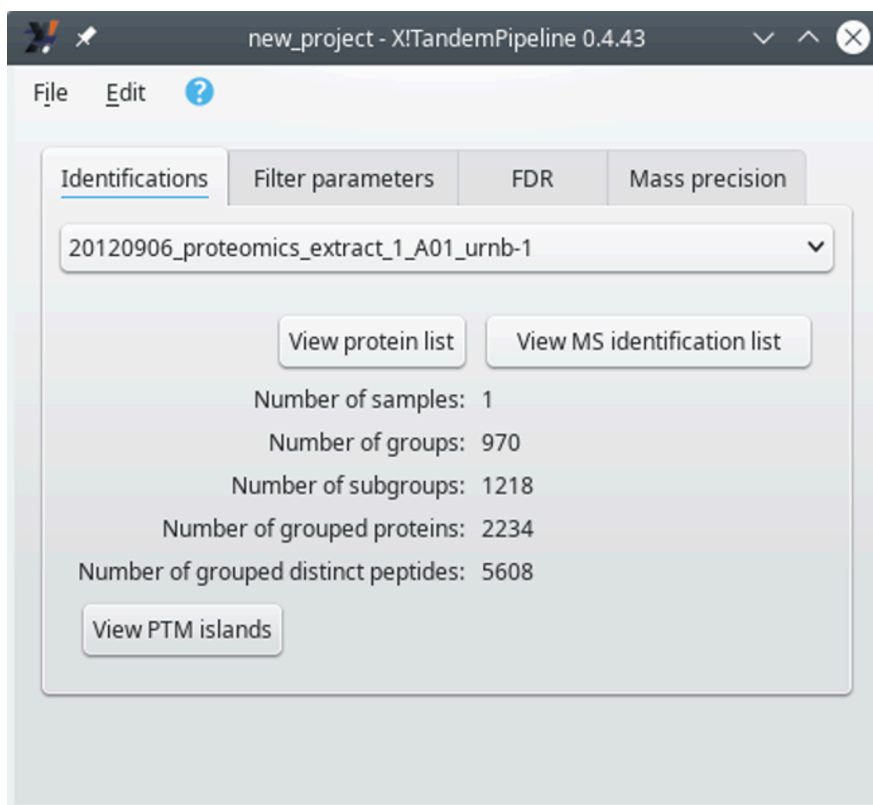
When loading multiple identification results files in *Individual* mode, the selection of any given identification results file is performed by selecting its name from the drop-down list widget *and* by clicking onto the *View protein list* button. Note that some metadata about the identifications are updated beneath the drop-down list widget.

FIGURE 3.8: SELECTING A PARTICULAR IDENTIFICATION RESULTS FILE'S DATA SET

Right after having selected an identification results file, click onto the *View protein list* to display the protein identifications list. That list has been obtained by performing the protein inference on the file's protein identification results (see Section 2.2.5.4, "Protein Inference: from PSMs to Protein Identities"). The window that opens up will be described later (see Section 4.1, "The Protein List Window").

💡 Tip

It is possible to open multiple protein list windows, each showing the identifications from a different file: maintain the `Ctrl` keyboard key pressed while clicking onto the *View protein list* button.

- *Choose results files*: by clicking onto the *Add files* button, the user is provided a file selection dialog window from which any number of protein identification results files might be selected for loading.

Note that it is possible to list all the opened protein identification results files by clicking onto the *View MS identification list* button. The window that opens up will be described later (see SECTION 3.4.2, "DISPLAYING THE MS IDENTIFICATIONS LIST").

- *Contaminants*: there are two possibilities here.

  - *Contaminants files*: when this radio button widget is selected, the list of contaminant proteins will be loaded from the files selected by clicking onto the *Add files* button.

  - *Contaminant regular expression*: when this radio button widget is selected, a text edit widget is shown, replacing the widget listing the contaminants database files. In this text edit widget, the user may enter a regular expression to match the accession number field of the protein databases that were used for the protein identification step. In this situation, the user must use specially crafted protein databases in which the contaminant proteins were tagged on the accession number using a particular text pattern. That particular text pattern is then matched against the *Contaminant regular expression* that the user enters in the text edit widget.

- *Contaminant removal mode*: there are two possibilities. The contaminant removal is the process by which, when identified proteins match proteins in the contaminants realm (either from the contaminants dabase files or as determined using the regular expression), they are disregarded for the later protein visualization steps.

  - *Protein list*: in this mode, as soon as a protein identification loaded from a protein identification results file matches a contaminant protein, it is disregarded.

  - *Groups*: in this mode, the protein inference process goes all the way through to the determination of the protein groups (see FIGURE 2.8, "PROTEIN INFERENCE: CONSTRUCTING A CONSOLIDATED PROTEIN IDENTIFICATIONS LIST"). When protein groups have sub-groups that contain a contamimant protein, then the whole group is disregarded. This might appear drastic, but our experience is that most often, the sub-groups in a group do identify proteins belonging to the same family. Therefore, if one protein is contaminant, all the other proteins in the group are supposed to be such also.

- *Peptide and protein filters*: this group box widget holds some parameters that configure the way protein inference is to be performed.

- *Peptide threshold on*: there are two possibilities:

  - *E-value*: all the PSMs having an expectation value higher than that value are disregarded. Enter the value in the spin box widget labelled *Peptide E-value*. A typical value for the *X!Tandem* engine is 0.05. When more stringent results are desirable, setting 0.02 should yield satisfactory results. See Section 2.2.5.2, "Computation of the Peptide Expectation Value (E-value)" of a detailed explanation of the E-value computation.

  - *FDR* (false rate discovery): the PSMs are disregarded if their FDR value does not match this parameter. Enter the value in the spin box widget labelled *Peptide FDR*. A typical setting is *1%*.

    ### Tip

    Using *FDR* is most useful when the identification results come from a database searching engine that does not compute an E-value. However, it does only work if the searching step was performed also on a decoy database. In *X!Tandem* the decoy database is crafted by reversing the peptide sequences. In this case, when proteins are identified on the basis of the reversed peptide PSM, then the protein identity is tagged with the "reversed" string, which might be used with the *Contaminant regular expression* setting defined earlier.

- *Number of peptides per protein*: this is the minimal required number of peptides that must be identified as belonging to a given protein in order to consider that protein identity as a valid one. These peptides have to be from non-contaminant proteins, of course.

- *Overall samples*: when checked and if multiple identification results files are to be loaded, then the *Number of peptides per protein* requirement might be fulfilled by looking for peptides in all the loaded files. For example, if one results file provides one peptide for a protein identification and another file provide another peptide (different from the first one) to identify the same protein, and if the *Number of peptides per protein* is 2, then the protein is considered as a valid protein. If not checked, that number of peptides requirement must be fulfilled by looking into each results file separately. This last setting is more stringent. A typical value for this setting is *2*.

Tip

This setting needs to be checked in at least one case: when a complex peptidic mixture is separated by ion chromatography (typically on an SCX—strong cation exchange—resin) and the different fractions are analyzed by bottom-up proteomics. The peptides coming from a given protein might be located in different fractions, and thus in different protein identification results files!

- *Protein Evalue*: threshold above which a protein identification is disregarded (see Section 2.2.5.3, "Computation of the Protein Expectation Value (E-value)").

- *Protein Evalue (log10)*: convenience spin box widget for the user to easily set the protein E-value.

- *Pep repro*: if set to 1, a peptide, to be accounted for, needs to be found in one protein identification results file. If set to a greater number, then that peptide needs to be found in that number of results files. This setting sets more stringent protein identification conditions each time it is incremented.

3.4.2   Displaying the MS Identifications List

This window displays a list of all the files that were involved in the *X!Tandem* run (first columns).

FIGURE 3.9: DISPLAYING THE MS IDENTIFICATIONS LIST (FIRST COLUMNS)

This window displays a list of all the files that were involved in the *X!Tandem* run (last columns).

FIGURE 3.10: DISPLAYING THE MS IDENTIFICATIONS LIST (LAST COLUMNS)

3.4.3 SAVING *i2MassChroQ* PROJECTS

Once exploration and optional modication of the identification data have been performed, the user can save the resulting data set into a *i2MassChroQ* project by selecting the *Save project* menu item of the *File* menu in the main program window (the extension of the file name typically should be `xpip`). See SECTION 3.5, "LOADING i2MASSCHROQ PROJECTS" for loading such a project.

3.5 LOADING *i2MassChroQ* PROJECTS

Loading of *i2MassChroQ* project files (file of `xpip`) extension) is only possible if the user has previously

- Loaded identification results;

- Saved the data to an *i2MassChroQ* project file using the *Save project* menu item of the *File* menu in the main program window.

4 Exploring identification data

This chapter describes in detail all the steps that the user accomplishes in their data exploration session. The general workflow is to start by looking at a protein identification results window and then by going into the details of the various identifications listed in it. This latter task entails looking into the peptides that provided the protein identification and then looking at the mass spectrum that provided the peptide identification. The mass spectrum, that is, the MS/MS spectrum, has features aimed at allowing the user to make an informed opinion on the validity of the peptide *vs* mass spectrum match (PSM) at hand. At each moment, it is possible to invalidate a PSM and the identification results are recomputed automatically by taking into account the modification entered by the user.

4.1 The Protein List Window

When identification results files are loaded, *i2MassChroQ* automatically performs the protein inference process by using the configuration settings described in Section 3.4.1, "Identification Data Loading Configuration".

4.1.1 The Protein List Table View

When the protein inference process is finished, *i2MassChroQ* displays the protein identifications list in a table view, as pictured in Figure 4.1, "The protein list window".

checked	group	accession	description	log(Evalue)	Evalue	spectra	specific spectra	sequences	specific sequences	coverage	MW	PAI	emPAI
✓	a1.a1.a1	GRMZM2G083841_P01	P04711 Phosphoenolpyruvate ...	-436.204	0	269	251	58	54	56.49 %	109272	1.84615	69.1704
✓	c117.a1.a1	GRMZM2G137839_P01	NP_001152746 ascorbate ...	-71.1121	7.72497e-72	32	7	9	2	52.00 %	27353.9	1.33333	20.5443
✓	c117.a2.a1	GRMZM2G054300_P01	NP_001150192 APx1 - Cytosolic ...	-59.8189	1.5174e-60	27	2	8	1	44.80 %	27290.8	1.16667	13.678
✓	c117.a2.a2	GRMZM2G054300_P04	NP_001150192 APx1 - Cytosolic ...	-59.8189	1.5174e-60	27	2	8	1	37.46 %	32330.3	1	9
✓		GRMZM2G054300_P02	NP_001150192 APx1 - Cytosolic ...	-42.1156	7.66366e-43	17		5		35.94 %	20841.5	1.125	12.3352
✓		GRMZM2G054300_P03	NP_001150192 APx1 - Cytosolic ...	-42.1156	7.66366e-43	17		5		31.80 %	23404.6	0.9	6.94328
✓	c407.a1.a1	GRMZM2G046841_P01	B6SIF9 Histone H2B ...	-24.4482	3.56257e-25	8	3	5	2	37.33 %	16133.9	0.714286	4.17947
✓	c407.a1.a2	GRMZM2G119071_P01	P30756 Histone H2B.2 ...	-24.4482	3.56257e-25	8	3	5	2	37.33 %	16163.9	0.714286	4.17947
✓	b36.a1.a1	GRMZM2G449496_P01	NP_001169327 hypothetical ...	-148.005	9.88939e-149	37	37	19	19	49.19 %	51820.9	0.88	6.58578
✓	b78.a1.a1	GRMZM2G027995_P01	Q41741 Eukaryotic initiation ...	-102.438	3.64897e-103	29	2	12	1	36.47 %	46952.9	0.9	6.94328
✓	b78.a1.a2	GRMZM2G027995_P02	Q41741 Eukaryotic initiation ...	-102.438	3.64897e-103	29	2	12	1	36.47 %	46952.9	0.9	6.94328
✓	b78.a2.a1	GRMZM2G116034_P02	NP_001104874 translation ...	-97.3364	4.60922e-98	29	2	12	1	36.47 %	46922.9	0.85	6.07946
✓		GRMZM2G116034_P01	NP_001104874 translation ...	-91.1493	7.09111e-92	28		11		33.58 %	46663.8	0.8	5.30957
✓	b60.a1.a1	GRMZM5G845611_P01	B4F8L7 Glyceraldehyde-3-...	-98.5797	2.63223e-99	52	36	14	11	37.64 %	47150.2	0.954545	8.00628
✓	b60.a2.a1	GRMZM2G337113_P02	P09315 Glyceraldehyde-3-...	-80.9409	1.14578e-81	41	25	10	7	33.25 %	42830	1.14286	12.895
✓		GRMZM2G129246_P01	NP_001146005 hypothetical ...	-73.2632	5.45513e-74	24		12		32.02 %	53278.8	0.576923	2.77505
✓	b82.a1.a1	GRMZM2G129246_P02	NP_001146005 hypothetical ...	-86.6197	2.40026e-87	25	25	13	13	46.07 %	40021.1	0.842105	5.95193
✓		GRMZM2G129246_P05	NP_001146005 hypothetical ...	-60.7567	1.75125e-61	20		10		40.45 %	32985.5	0.8125	5.49382
✓		GRMZM2G129246_P03	NP_001146005 hypothetical ...	-34.8504	1.41117e-35	10		7		38.89 %	24162.3	0.538462	2.45511
✓		GRMZM2G129246_P04	NP_001146005 hypothetical ...	-34.8504	1.41117e-35	10		7		38.89 %	24162.3	0.538462	2.45511
✓	c117.a3.a1	GRMZM2G140667_P01	NP_001105500 ascorbate ...	-24.1051	7.85039e-25	10	8	5	4	24.13 %	30900.7	0.357143	1.27585
✓	c117.a3.a2	GRMZM2G140667_P02	NP_001105500 ascorbate ...	-24.1051	7.85039e-25	10	8	5	4	24.47 %	30482.5	0.384615	1.42446
✓		GRMZM2G140667_P04	NP_001105500 ascorbate ...	-20.5201	3.01938e-21	8		4		31.41 %	20781.4	0.5	2.16228
✓		GRMZM2G175867_P01	NP_001130422 hypothetical ...	-1.60206	0.025	1		1		2.12 %	66514.1	0.0454545	0.110336
✓		GRMZM2G175867_P02	NP_001130422 hypothetical ...	-1.60206	0.025	1		1		3.61 %	38959.6	0.0588235	0.145048
✓		GRMZM2G153969_P01	NP_001149564 OB-fold nucleic ...	-2.52288	0.003	2		1		8.43 %	18248.5	0.142857	0.389495
✓		GRMZM2G174479_P01	B6TM56 Chloroplast outer ...	-3.85387	0.00014	2		1		11.40 %	12394.1	0.166667	0.467799
✓		GRMZM2G167698_P01	seq=translation; ...	-2.5376	0.0029	2		1		2.48 %	58821.2	0.04	0.0964782
✓		GRMZM2G009232_P01	NP_001141257 hypothetical ...	-2.5376	0.0029	2		1		2.49 %	58693.1	0.0416667	0.100694
✓		GRMZM2G009232_P02	NP_001141257 hypothetical ...	-2.5376	0.0029	2		1		3.95 %	37105.2	0.0769231	0.193777
✓		GRMZM2G009232_P03	NP_001141257 hypothetical ...	-2.5376	0.0029	2		1		3.94 %	37454.5	0.0769231	0.193777
✓	c141.a1.a1	GRMZM2G318635_P01	seq=translation; ...	-66.2764	5.29139e-67	16	16	10	10	9.69 %	190929	0.136986	0.370839
✓	c530.a1.a1	GRMZM2G157462_P01	NP_001152484 dynamin-2A ...	-20.5331	2.93026e-21	7	7	4	4	4.93 %	99589.6	0.0851064	0.216484

search accession ▾

proteins all:6814 valid:3895 valid&checked:3895 grouped:2481 displayed:6814

The protein identifications list window displays the proteins assembled into groups. A number of metadata about the identifications are shown in a number of colums, the contents of all of which are described in detail in the text.

FIGURE 4.1: **THE PROTEIN LIST WINDOW**

The columns that make the protein list table view are detailed below:

- *Checked*: if checked, the identified protein listed on the table row is set to an "accepted" state. By default, all proteins are set to this accepted state. Unchecking a protein determines the protein inference reprocessing, because disregarding a protein modifies the whole protein identifications results set;

- *group*: the group the protein belongs to;

- *accession*: the accession number field of the protein database;

- *description*: the description field in the protein database;

- *log(E-value)*: the Log10 of the protein E-value;

- *E-value*: the protein E-value;

- *spectra*: the number of spectra that identified the protein;

- *specific spectra*: the number of spectra that identified *only* this protein;

- *sequences*: the number of peptidic sequences that can be assigned to this protein;

- *specific sequences*: the number of peptidic sequences that can be assigned *only* to this protein;

- *coverage*: the percentage of the protein sequence covered by the peptides that identified it;

- *MW*: the molecular weight of the protein (M_r);

- PAI: "Protein abundance index". This index was defined as the "number of peptides identified divided by the number of theoretically observable tryptic peptides". See HTTPS://WWW.NCBI.NLM.NIH.GOV/PMC/ ARTICLES/PMC186633/ ;

- emPAI: "Exponentially modified protein abundance index". This index was defined as emPAI = $10^{PAI} - 1$. See HTTPS://PUBMED.NCBI.NLM.NIH.GOV/15958392/ .

It is possible to select the columns that must be displayed in the table by checking or unchecking the corresponding item in the *Columns* menu.

The *Show only* menu allows one to select the kind of protein items to be shown:

- *Valid proteins*: when checked, the program only shows valid proteins, that is, protein identifications that fullfill the restriction parameters, like protein E-value, for example. These parameters were set at protein identification results loading time but can be modified later;

- *Checked proteins*: show only the proteins that were checked. This setting is useful when the user has unchecked a number of proteins and that they want to regularly keep an eye on them. When proteins are unchecked, the protein inference process is run anew to compute a new grouping by taking *not* into account the proteins that were disregarded;

- *Grouped proteins*: only show the proteins that belong to a group.

The protein identifications list table view above shows greyed protein identities. These are proteins that, by current filter parameters (E-value threshold, for example), are considered *not* valid.

4.1.2   OPERATIONS IN THE PROTEIN LIST WINDOW

The *Protein list* window houses a number of useful features that let the user scrutinize the protein identifications and also modify the results to suit either more or less stringent filtering parameters.

Searching data in the table view.   One interesting feature of the *Protein list* window is the ability to search through the table's contents using the *Search* item at the bottom of the window. A number of fields of the protein record, that is, columns in the table view, might be searched.

Dynamic setting of the filter parameters. *i2MassChroQ* provides a rather high level of flexibility: once a protein identification results set of files has been loaded and that the protein inference process is achieved, the resulting protein groups are displayed in the *Protein list* window. At this time, the grouping was performed using the parameters set as pictured in Section 3.4.1, "Identification Data Loading Configuration". It is nonetheless possible to modify these parameters on the fly using the main program window's *Filter parameters* tab, as pictured in Figure 4.2, "Protein identification filter parameters tab of the main window".
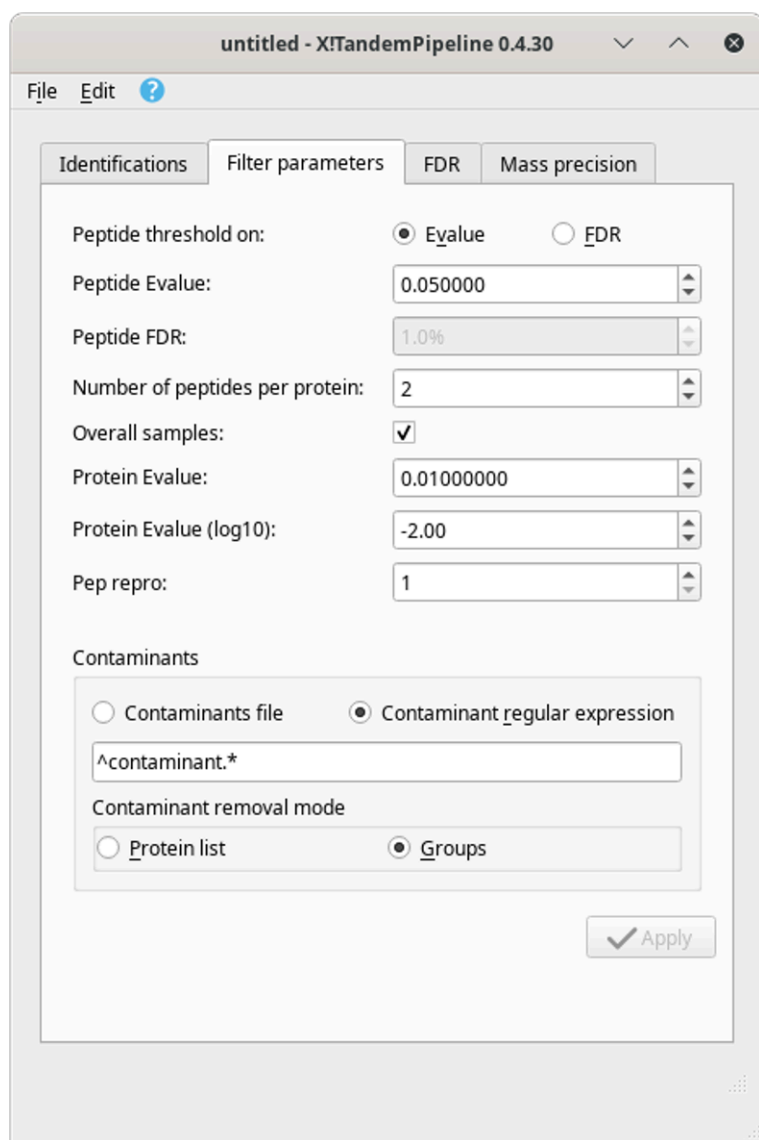
The filter parameters in this dialog box window do mirror the ones that one can set prior to loading protein identification results files. When modified, these parameters elicit a complete run of the protein inference process.

FIGURE 4.2: **PROTEIN IDENTIFICATION FILTER PARAMETERS TAB OF THE MAIN WINDOW**

Real time update of the false discovery rate. The false discovery rate (FDR) is recalculated at each protein inference process. The data regarding this quality assessment criterion are shown in Figure 4.3, "False discovery rate (FDR) data after a protein inference process is run".
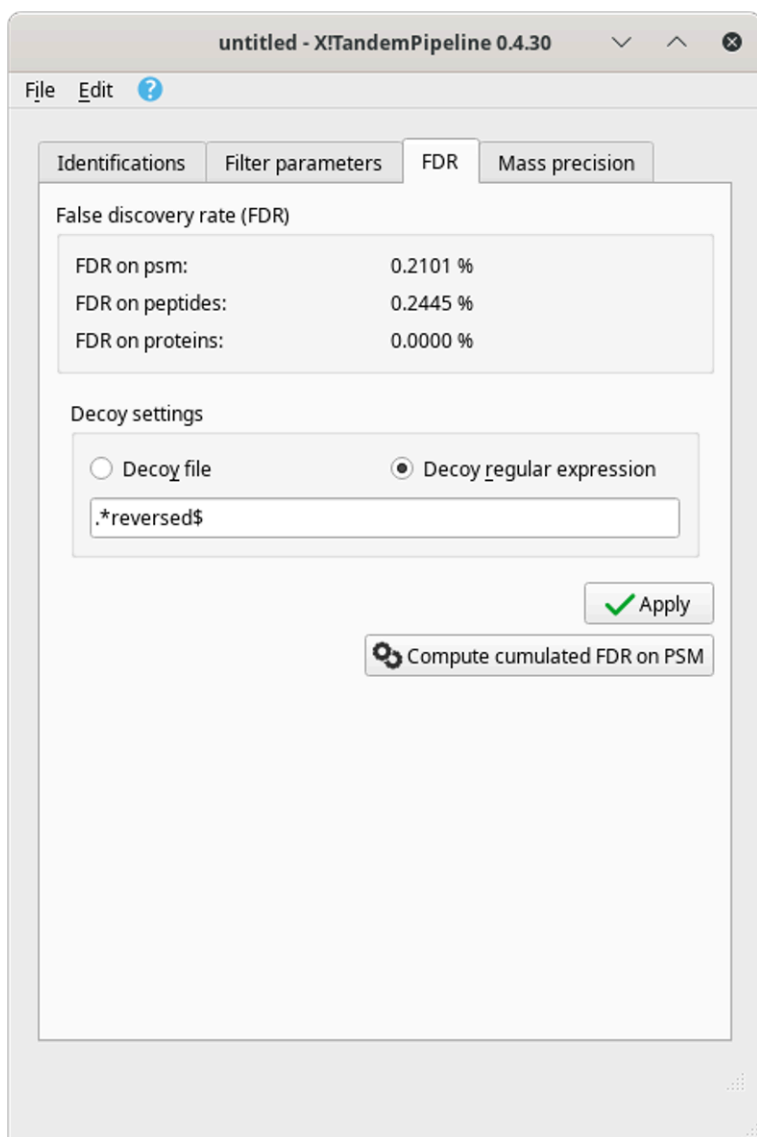
The various data bits about the false discovery rate that is computed each time a protein inference process is run. Note that it is possible to modify the *Decoy settings*, after which the *Apply* button triggers the recalculation of the FDR.

FIGURE 4.3: **FALSE DISCOVERY RATE (FDR) DATA AFTER A PROTEIN INFERENCE PROCESS IS RUN**

Distribution of mass errors on PSMs plotted in a histogram. It is possible to visualize the distribution of the mass errors over the whole dataset, as pictured in FIGURE 4.4, "MASS PRECISION QUALITY ASSESSMENT". The histogram plots the number of mass spectra that could achieve a PSM against the mass error (mass delta), that is, the difference between the experimental peptide mass and the calculated peptide mass. FIGURE 4.4, "MASS PRECISION QUALITY ASSESSMENT".
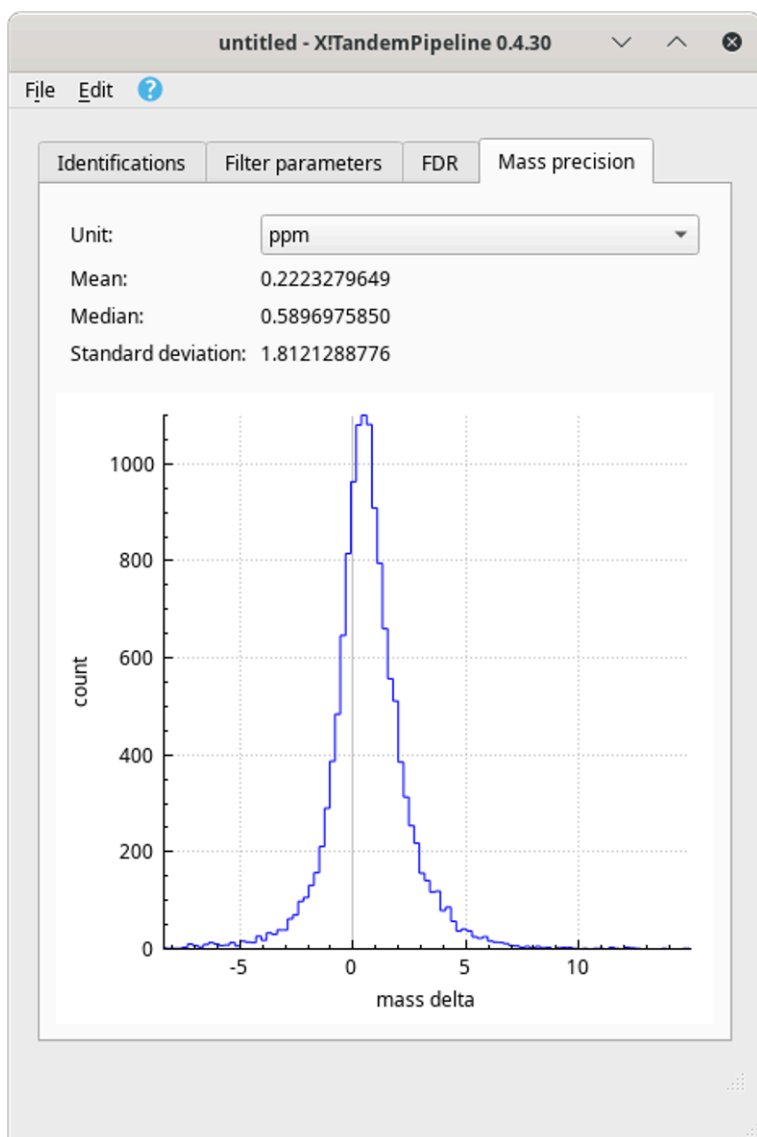
The histogram plots the number of PSMs against the mass error calculated between the experimental mass of the peptide and the calculated mass.

FIGURE 4.4: MASS PRECISION QUALITY ASSESSMENT
The mass delta calculation involves only the peptides that successfully identifed proteins that are currently checked in the protein identification list and that satisfy the filter parameters. The proteins identified in the decoy database are not processed. The unit of the mass delta may be selected using the *Unit* drop-down list. Two units are available: ppm (for part-per-million) or Dalton.

Exporting the final protein identifications list to a spread sheet. Once all the proteins in the identifications list have been properly checked, the user might export the data set to an OpenDocumentFormat (ODF) spread sheet file using the *As ODS file* menu item of the main window's *Export* menu.

4.1.3 Delving Inside the Protein Identification Data

The protein list table view, as pictured in Figure 4.1, "The protein list window" is actually an active matrix in which the user can easily trigger the exposition of the data that yielded any protein identification element of the table. This is simply done by clicking onto any cell of the table at the row matching the protein for which scrutiny of the data is desired.

Depending on the column at which the mouse click happens, there might be two different windows showing up:

- The *Protein details* window, showing the sequence of the protein, the matching peptides and other informational data bits, as pictured below:
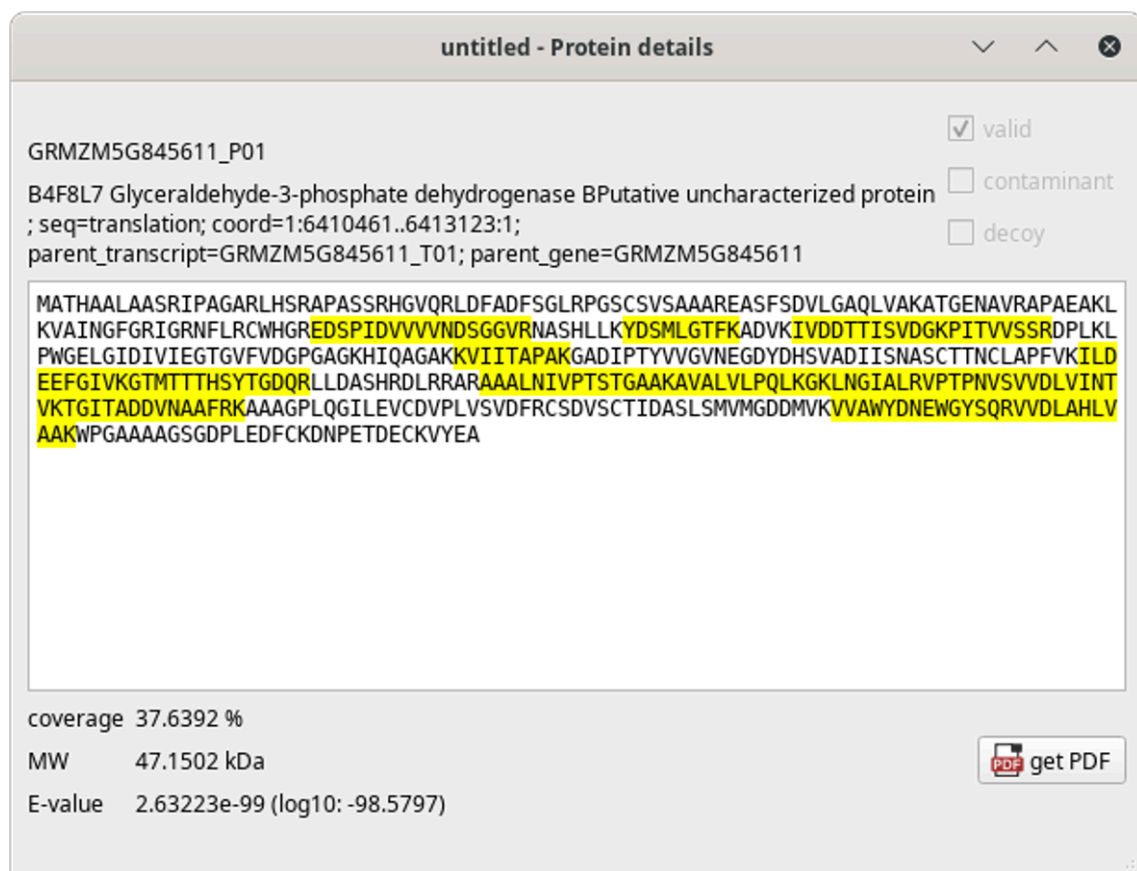
When one cell in the *Accession*, *Description* or *Coverage* column is clicked, this window shows up and displays the sequence of the protein, the coverage of the peptides and other useful data.

FIGURE 4.5: PROTEIN DETAILS WINDOW

- When one cell in any one of the remaining columns is clicked, the window that shows up is the *Peptide list* window showing a list of all the peptide identifications, to be described in the next section.

Tip

When clicking one cell in one column and one given row, the corresponding window shows up, if one was not already open. If one window is already open, no other window shows up, but the existing window has its data updated to match the new protein row being clicked on.

It is possible to have multiple windows opened at a time by clicking a new row while maintaining the `Ctrl` key pressed.

4.2 The Peptide List Window

The *Peptide list* window displays all the data in a table view similar to the one used to display the protein list described in the previous sections.

4.2.1 The Peptide List Table View

The *Peptide list* table view has a pretty large number of columns to display all the data about each peptide that identified a given protein. These columns are described in the following figures.

The *Peptide list* table view has many columns (first columns).

FIGURE 4.6: THE PEPTIDE LIST WINDOW (FIRST COLUMNS)

The *Peptide list* table view has many columns (last columns).

FIGURE 4.7: **PEPTIDE LIST WINDOW (LAST COLUMNS)**

The table's contents are well described by the column headers that are self-explanatory. When hovering over a column header with the mouse cursor, a tool-tip explanatory text is displayed.

It must be noted that more columns might make the table view depending on the protein identification data that were loaded. Indeed, depending on the database searching engine that was used for the protein identification, the data to be displayed vary. The whole list of columns that might be displayed in the table view are pictured in

FIGURE 4.8, "COLUMNS THAT POPULATE THE PEPTIDE LIST TABLE VIEW"

Depending on the provenience of the protein identifications (the database search engine), the columns that are part of the table view differ. This full list is displayed when selecting the *Columns* menu.

FIGURE 4.8: **COLUMNS THAT POPULATE THE PEPTIDE LIST TABLE VIEW**

4.2.2 Operations in the Peptide List Window

The *Peptide list* window houses a number of pretty interesting features that let the user scrutinize the peptide details.

Searching data in the table view.   One interesting feature of the *Peptide list* window is the ability to search through the table's contents using the *Search* item at the bottom of the window. A number of fields of the protein record, that is, columns in the table view might be searched.

Exporting the final protein identifications list to a spread sheet.   Once all the peptides in the identifications list have been properly checked, the user might export the data set to an OpenDocumentFormat (ODF) spread sheet file using the *As ODS file* menu item of the main window's *Export* menu.

4.2.3 Delving Inside the Peptide Identification Data

The *Peptide list* table view, as pictured in Figure 4.6, "The peptide list window (first columns)" is actually an active matrix in which the user can easily trigger the exposition of the data that yielded any peptide identification element of the table. This is simply done by clicking onto any cell of the table at the row matching the peptide for which scrutiny of the data is desired.

4.2.3.1 The Peptide Details Window

When clicking any one of the cells of the peptide list table view, one window shows up that details the various data elements for the peptide documented in the table row. The window is pictured in Figure 4.9, "Peptide details window".
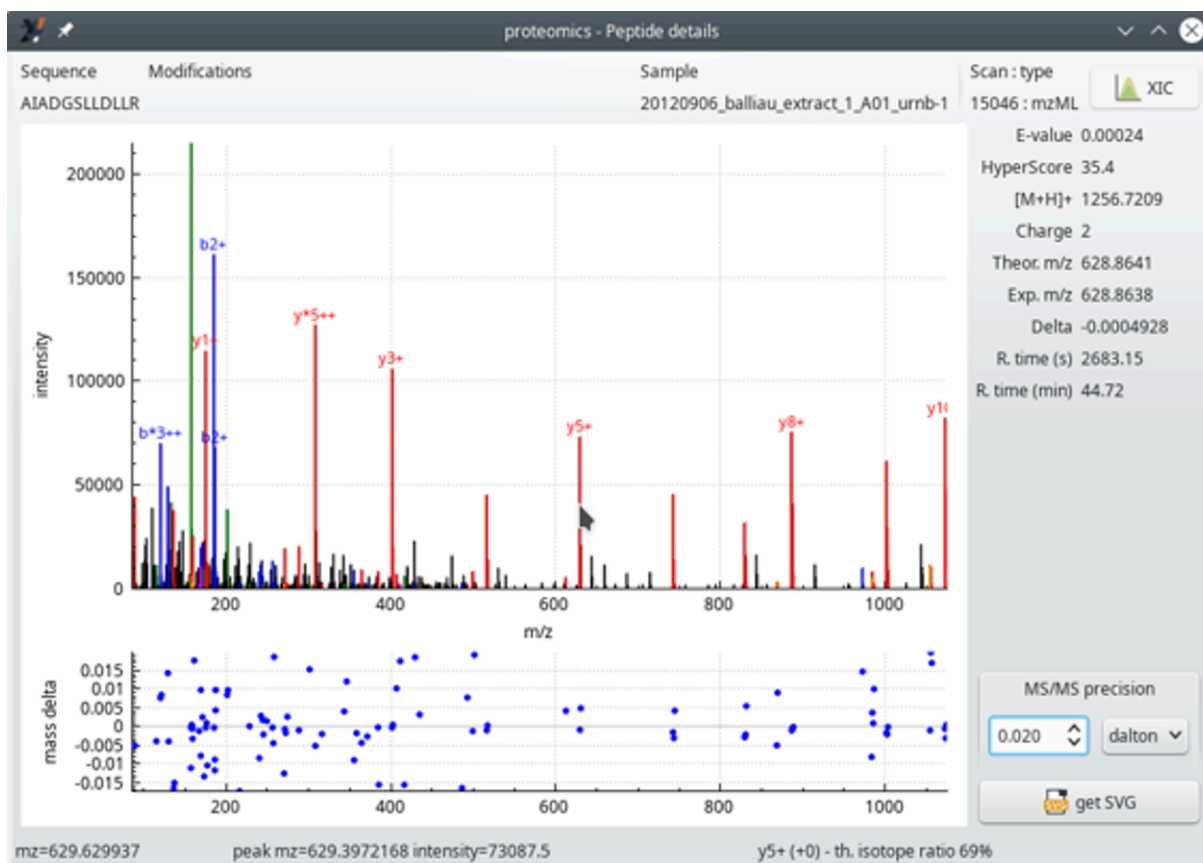
This window displays the MS/MS spectrum that allowed identifying a peptide (that is, a PSM). A number of informational data bits are displayed, like the MS/MS scan number, the E-Value for the peptide, along with its Hyperscore, for example (see text below for a thorough description).

FIGURE 4.9: PEPTIDE DETAILS WINDOW

In Figure 4.9, "Peptide details window", the two graphs show the following:

- The top graph displays the mass spectrum of this PSM. This MS/MS spectrum has its recognized peaks in the *b* and *y* ion series labelled in blue and red respectively. When the mouse cursor hovers over a mass peak, the details of that mass peak are printed in the status bar of the window (bottom line).
  Navigating the spectrum is straightforward: to zoom/unzoom in a given area of the spectrum, point the mouse cursor at the peak of interest and use the mouse wheel to zoom/unzoom. To modify the ordinate intensity scale, click onto the axis and drag the mouse upwards or downwards.

- The bottom graph plots—for each matching MS/MS peak (that is, *b* and *y* ion series)—the mass difference (mass delta) between the ion's measured mass and the theoretical mass. In this example, we see that the *y* ion series is moderately matched (large error range).
  It is possible to set the *MS/MS precision* to a determinate value and unit (Dalton, ppm or res). The value entered in the spin box widget modifies the assignement of the fragmentation peaks.

Tip

The MS/MS spectrum mass peaks are annotated using the following naming convention:

- *: neutral NH_3 loss;

- *o*: neutral H_2O loss;

The ion charge is displayed in the form of "+" or "++" text strings.

The right hand side margin of the window provides a number of data about the PSM, like the peptide E-value, the HyperScore, the ion charge, the theoretical and experimental masses, the difference between the two, the retention time at which this ion was detected… These informational data bits are self-explanatory.

The *XIC* button at the top right corner of the window triggers the calculation of the extracted ion current chromatogram, as described in the section below.

4.2.3.2 The XIC Viewer Window for the Peptide Details

One interesting feature of the *Peptide details* window, is the *XIC* button (top right) that triggers the calculation of an extracted ion current chromatogram, as pictured in Figure 4.10, "The extracted ion current (XIC) chromatogram viewer window".

## Tip: What is a XIC chromatogram?

The notion of *extracted ion current* chromatogram is best explained by describing the computation that yields that chromatogram.

The user defines the m/z value for which the chromatogram is to be determined. The program iterates in each MS spectrum (that is, full scan spectrum) and looks if an ion by that m/z value was encountered. If so, a variable holding the cumulated intensity of that ion is incremented for the retention time at which the mass spectrum was acquired. For example, if m/z value 1254.25 is searched for, and an ion of that m/z value is found in the mass spectrum acquired at retention time 2.5 min, then a tuple variable is stored like this: (2.5, intensity). Then, another mass peak by that m/z value is found in mass spectrum acquired at retention time 47 min, for which another tuple is created: (47, intensity).

If the data are from ion mobility—mass spectrometry (IM-MS) experiments, there might be a large number of spectra acquired at a given retention time. For example, data from the *Waters Synapt2*™ instrument have 200 spectra acquired for any given retention time value (the spectra are drift-related spectra). In *Bruker timsTOF*™ data, there are more than 700 spectra acquired at any given retention time. Thus,

the searched m/z value might be found more than once for a retention time value. In this case, the tuple's intensity value is incremented by the intensity of the new peak of the m/z value at that specific retention time value.

When the program has finished iterating in all the mass spectra of the acquisition, it plots the XIC chromatogram as intensity=f(retention time). This is the reason why it is considered a chromatogram.
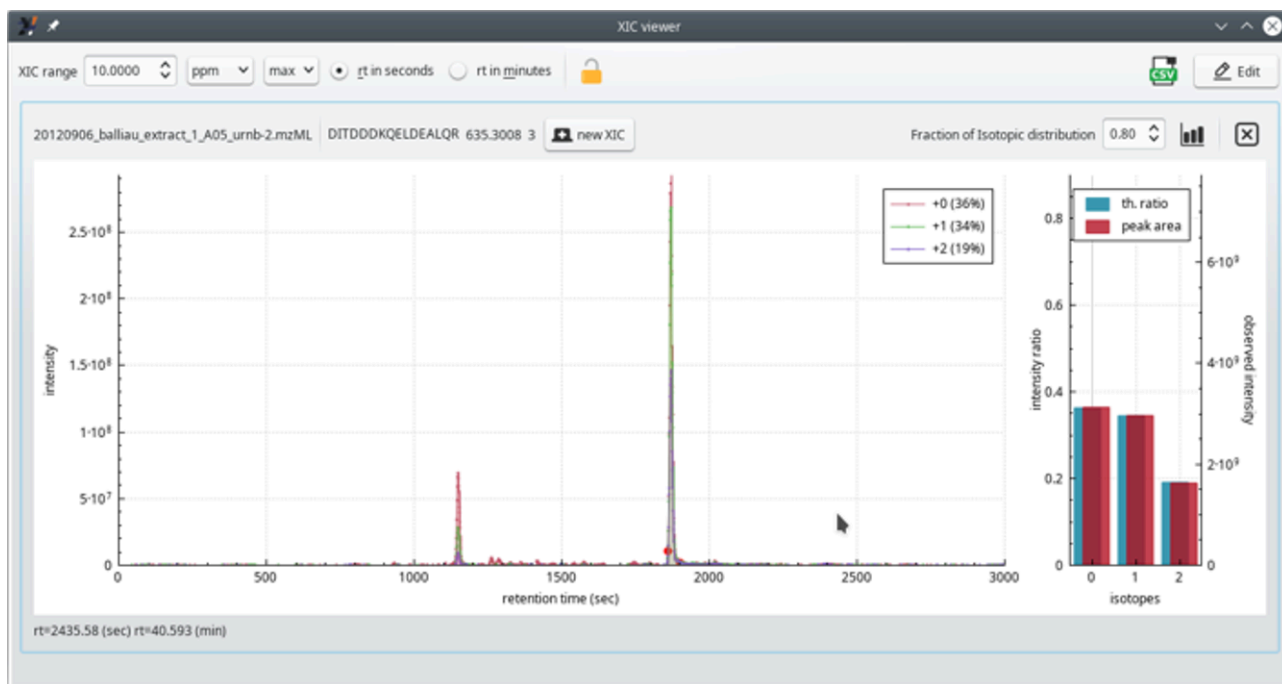
The extracted ion current (XIC) chromatogram viewer is useful to scrutinize the mass data at the very origin of a PSM. It is routinely used to ensure that the PSM is faithful. If not, the corresponding peptide can be unchecked from the peptide identifications list table view. As a response, the protein inference process is run anew.
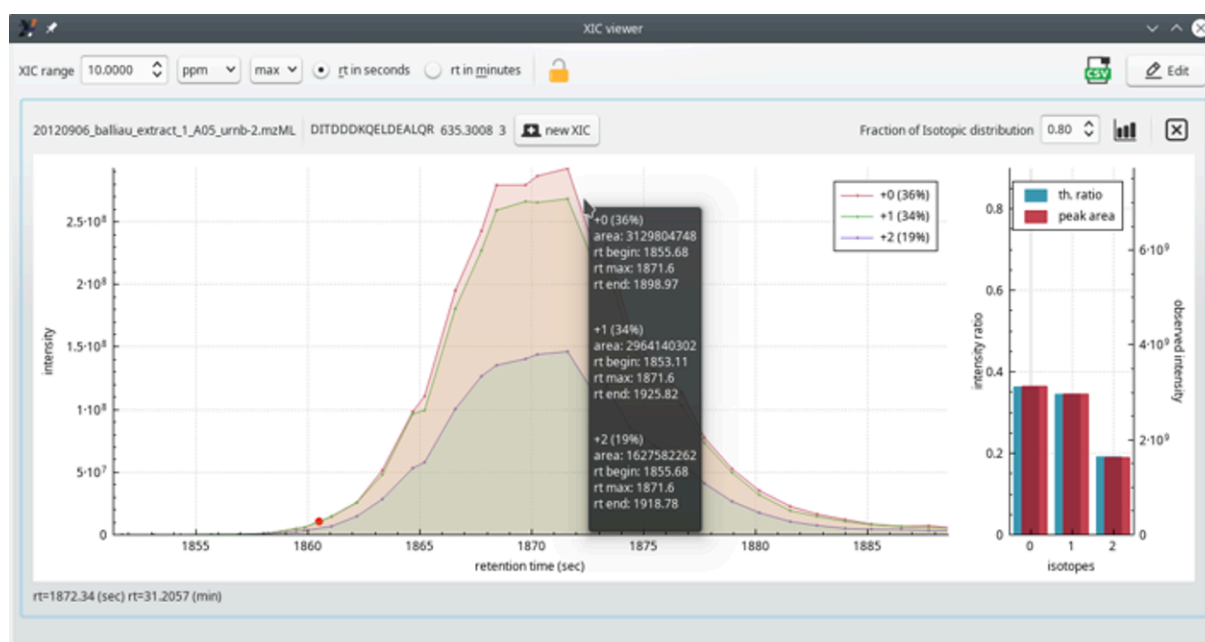
FIGURE 4.10: **THE EXTRACTED ION CURRENT (XIC) CHROMATOGRAM VIEWER WINDOW**

The *XIC viewer* window displays the "guts" of the of MS spectrum of the precursor ion that was fragmented and that yielded a PSM. The XIC chromatogram (left plot panel) is actually a set of XIC chromatograms that are superimposed in the plot widget (see FIGURE 4.11, "THE EXTRACTED ION CURRENT (XIC) CHROMATOGRAM VIEWER WINDOW (ZOOMED VIEW)"). One of the traces (legend +0) is for the first peak of the isotopic cluster of the searched ion; the second trace (legend +1) is for the second peak of the isotopic cluster. Likewise for the third trace. In the typical informatics-oriented style of numbering, the first isotopic peak (only light isotopes enter in the composition of the peptidic ion), is "isotope 0"; the second isotopic peak (one light isotope is substituted with a heavy one) is "isotope 1" and, finally, the third isotopic peak (two light isotopes were replaced by heavy ones) is "isotope 2".

The right panel is a bar plot showing —for each one of the isotopes— a comparison between the experimental peak area and the computed probability of the corresponding isotope peak. In the example, the match between the experimental and the theoretical cluster shape is perfect. This scrutinization of the data is very useful when one wants to double-check the quality of a protein identification on the basis of a given PSM.

✎ NOTE

The theoretical isotopic cluster peaks are calculated using the formula of the peptide that has been identified in the PSM for which the XIC chromatogram is being requested.

Zoomed in view of the three XIC chromatogram plots in the left hand side plot widget.

FIGURE 4.11: **THE EXTRACTED ION CURRENT (XIC) CHROMATOGRAM VIEWER WINDOW (ZOOMED VIEW)**

Another interesting bit of information is the *Fraction of Isotopic distribution* (spin box widget top right corner of the window). This one needs some background. When one has a peptide formula and the peptidic ion charge, one can calculate the theoretical isotopic cluster corresponding to that specific ion. The calculation is CPU-intensive and sometimes one would like to limit its duration. This is possible by indicating that one is interested, for example, in only the 80 % of the total isotopic peaks that one would effectively find (even in minute amouts) in nature. This value tells exactly that. The calculation displayed in the window, encompassing only 80 % of the whole natural span of the isotopic cluster, yields a calculated cluster made of only three isotopic peaks. If the user had set the value to 99 %, then, most probably, numerous other isotopic peaks of very low intensity would have been calculated on the right hand side of the isotopic cluster (heavier ions because more heavy isotopes are included in the computation).

Remember to click onto the "*Histogram plot*" button next to the spin box widget for the new *Fraction of Isotopic distribution* value to take effect.

As for the previous MS/MS spectrum plot, to zoom in/out regions of the XIC chromatogram plot widget, hover the mouse cursor over the region of interest and rotate the mouse wheel.

4.3 HANDLING PHOSPHO-PROTEOMICS DATA

i2MassChroQ is able to cope with phospho-peptides. The mass spectrometric data are acquired exactly as usual with the mass spectrometer, but the sample preparation goes along theses steps:

- Separate digestion of the samples (when there are more than one);

- Labeling of the peptides, each sample gets a different label;

- Pool of the whole set of peptides into a single mixture;

- Separation of the peptides on a strong cation exchange (SCX) resin, collection of the fractions;

- Phospho-peptide enrichment using IMAC[1] for each SCX fraction. The SCX fraction is loaded onto the IMAC resin and, following a wash step, the phospho-peptides are eluted (pH-based elution). There is thus a one-to-one relation between a SCX fraction and an IMAC-based purification fraction.

- Mass spectrometric analysis of each IMAC-based phospho-peptide-enriched fraction.

X!Tandem needs to be configured in such a manner that it can generate all the theoretical peptides (and fragments) that might bear the phosphoryl group. This process is described in the section below.

---

1 Immobilized-metal affinity chromatography.

5 Exploring post-translational modification data

This chapter describes in detail all the steps that the user accomplishes in their post-translational modification (PTM) data exploration session.

5.1 Setting the *X!Tandem* Run Presets for Phospho-proteomics

The very first step in starting a phospho-proteomics-based protein identification run is to configure the run so that the database search engine can model phopshorylated peptides on the basis of the sequence of the peptides in the database. That configuration step is started as described in Figure 5.1, "X!Tandem settings window for a phospho-proteomics project".
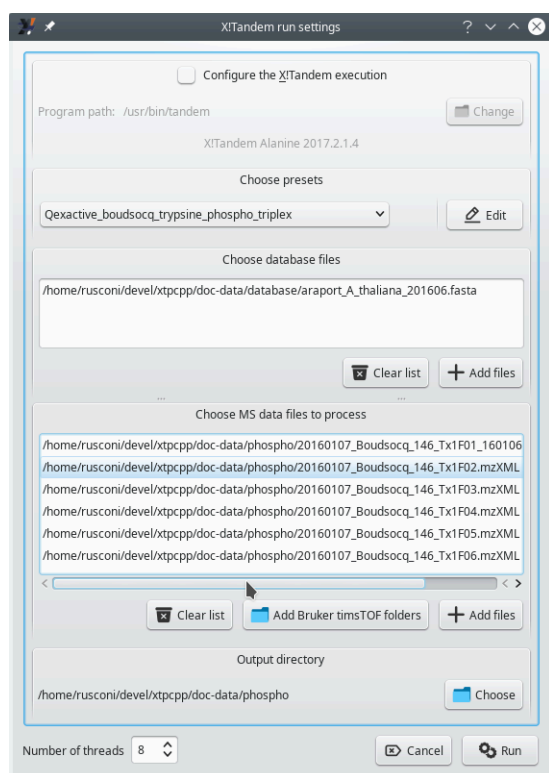

The *Choose presets* option in this window allows the user to select an *X!Tandem* presets file to suit the protein identification run. In this example, the chosen presets file contains a number of configuration bits specific for a phospho-proteomics project.

FIGURE 5.1: *X!TANDEM* SETTINGS WINDOW FOR A PHOSPHO-PROTEOMICS PROJECT

The configuration of *X!Tandem* needs to be performed by using the presets method, described in Section 3.3, "Setting the X!Tandem Run Presets" and following sections. The *Edit* button next to the drop down list allows one to edit the presets that configure the handling of the database by the *X!Tandem* database search en-

gine. The window that opens up upon clicking onto that button has two tabs that require the user's attention, as shown in Figure 5.2, "Setting the project to be a phospho-proteomics project" and Figure 5.3, "Configuring the phosphorylated residues".

As described in Section 3.3.1, "Loading existing presets configurations from file", it is possible to load existing presets in case these were defined already and might be reused for a repeated data exploration session on the same data set.
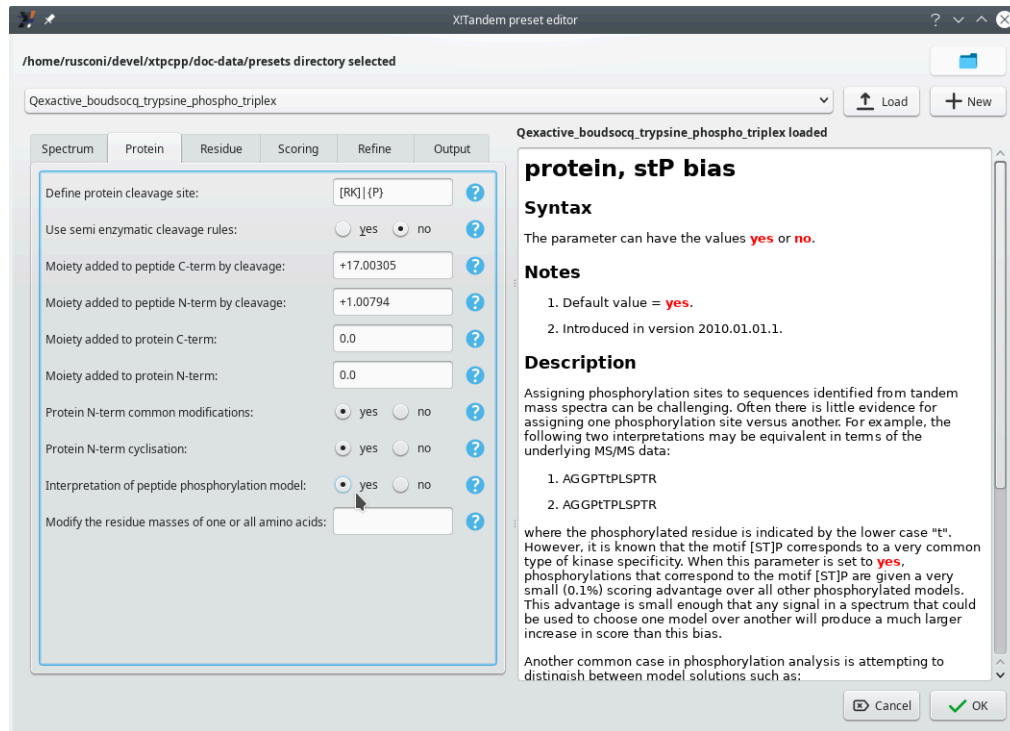
The major bit that need the user's attention in the *Preset editor* window's *Protein* tab is the *Interpretation of peptide phosphorylation model* to *yes*. The question mark icon on the side of that configuration option displays explanatory text on the right hand side of the window.

FIGURE 5.2: SETTING THE PROJECT TO BE A PHOSPHO-PROTEOMICS PROJECT
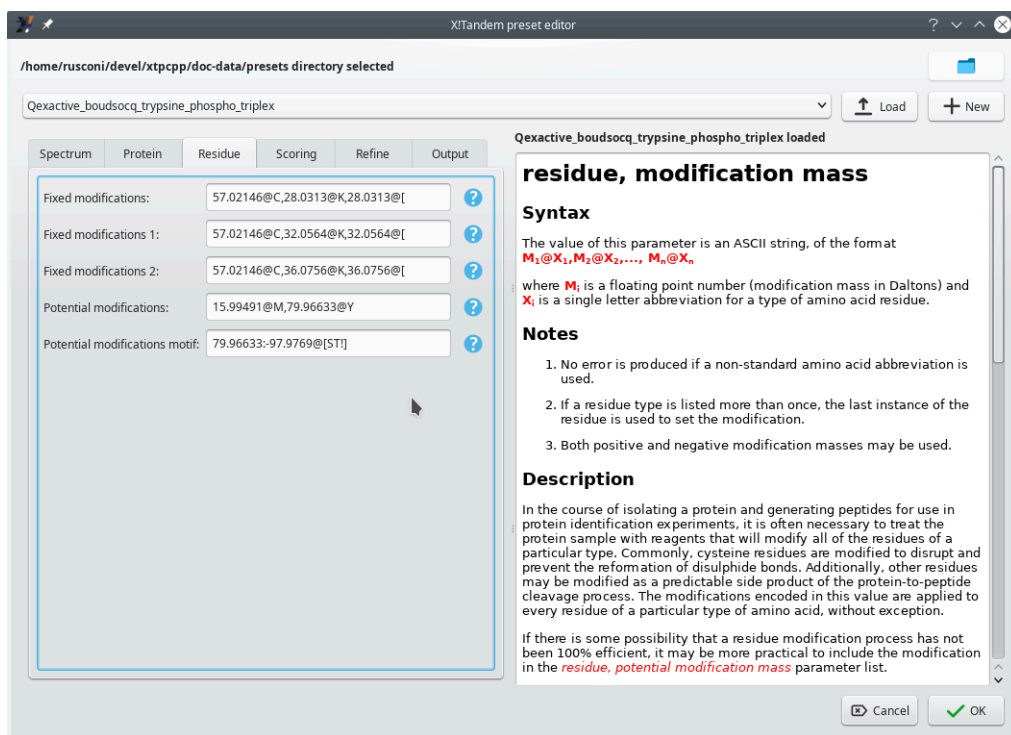
The phosphorylation events are not *Fixed modifications* because it is not possible to predict that they will occur systematically. This is the reason why the phosphorylation events are configured as *Potential modifications*.

FIGURE 5.3: CONFIGURING THE PHOSPHORYLATED RESIDUES

FIGURE 5.3, "CONFIGURING THE PHOSPHORYLATED RESIDUES" shows how to configure the potential phosphorylation of selected residues (tyrosine, threonine and serine). Setting the *X!Tandem* parameters for phosphoproteomics analyses involves specifying the mass difference between unmodified and modified residues and the nature of the residue that might bear the modification. There are two different settings available:

- *Potential modifications*: the *Y* residue might be phosphorylated, with a net mass increment of *79.96633* Da.

- *Potential modifications motif*: the *S* and *T* residues might be phosphorylated (net mass increment of *79.96633* Da) or be subject to a neutral phosphoric acid loss (net mass loss of *97.9769* Da).

The reason why the potential phosphorylation of tyrosine (Y) is not mentioned along with the S and T modifications in the motif setting is that phosphorylated tyrosine residues do not suffer from phosphoric acid neutral loss upon collisionally activated dissociation (CID). Phosphorylated serine and threonine residues are readily dephosphorylated upon CID. Hence, the requirement to configure both the phosphorylation and the dephosphorylation events as a PROSITE motif (see the question mark help).

⚡ Tɪᴘ

The loss of a phosphoric acid molecule *(not ion)* is called a "neutral loss". By essence, the lost molecule cannot be detected, because it bears no charge. However, the search software may detect that there might be a negative mass delta between calculated fragments bearing a phoshoryl group and the measured mass of product ions. In this eventuality, the software may deduce that the fragment was phosphorylated before the fragmentation occurred.

The mass spectrometer might be configured to monitor neutral phosphoric acid loss, or not. In some instruments, that workflow is not available; however, in these instruments a higher energy collisional dissociation[1] process elicits two fragmentation events: loss of a phosphoric acid molecule and peptide backbone dissociation. In this case, the database searching engine (*X!Tandem*, for us) is instructed to monitor the loss of phosphoric acid (that is, a neutral loss) on the product ions of the y ion series. In the best cases (best sequence coverage by the product ions), it is thus possible to locate the phosphoryl group on the peptide.

❗ Cᴀᴜᴛɪᴏɴ: Pʜᴏsᴘʜᴏ-ᴘʀᴏᴛᴇᴏᴍɪᴄs ᴘʀᴏᴊᴇᴄᴛs ᴏғᴛᴇɴ ɪɴᴠᴏʟᴠᴇ ʟᴀʙᴇʟ-ʙᴀsᴇᴅ ǫᴜᴀɴᴛɪғɪᴄᴀᴛɪᴏɴ

The *Residue* tab of the *X!Tandem preset editor* window shown above lists a number of fixed modifications that need an explanation, because we'll find them later in other figures. The *57.02146@C* modification is the carbamidomethylation of cysteine residues. The various 28, 32 and 36 mass increments on lysine residues are the di-methylation modifications with (32, 36) or without (28) heavy isotopes[2]. The exact same mass increments labelled with the "@[" notation are the same modifications occurring on the N-terminus of the peptide.

Once all the settings have been validated, click the *Run* button, in the same manner as described at Sᴇᴄᴛɪᴏɴ 3.3.4, "Rᴜɴɴɪɴɢ ᴀ ᴘʀᴏᴘᴇʀʟʏ ᴄᴏɴғɪɢᴜʀᴇᴅ X!Tᴀɴᴅᴇᴍ ᴘʀᴏᴄᴇss", to actually start the database search process.

---

1 In Orbitrap analyzer-based instrument, HCD stands for "higher-energy C-trap dissociation". However, a more generic term is oft-used: "higher energy collisional dissociation".

2 Boersema *et al.* 2009. Multiplex peptide stable isotope dimethyl labeling for quantitative proteomics. *Nature Protocols.*

5.2   Loading the Protein Identification Results

The loading of the protein identification results for a phospho-proteomics project occurs in an identical fashion as for a non phospho-proteomics project, as described in Section 3.4, "Loading the Protein Identification Results". The main program window shows the same *Identifications* tab as described before (see Figure 5.4, "The main window's Identifications tab").
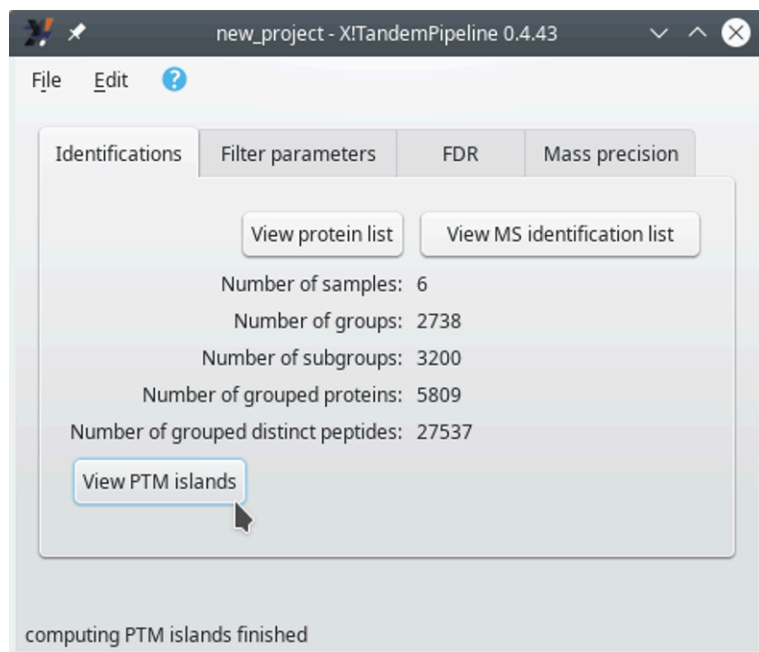
The *Identifications* tab inside of the main program window after loading a PTMs-based project's identification results files. The *View protein list* and *View MS identification list* buttons perform exactly as described earlier. In order to trigger the PTMs-based data exploration session, click the *View PTM islands* button.

FIGURE 5.4: THE MAIN WINDOW'S *IDENTIFICATIONS* TAB

When *i2MassChroQ* has done loading all the results, the *Protein list* window opens up as it usually does. This window, however, is not devoted to the exploration of phosho-proteomics data. In order to start the exploration of phospho-proteomics data, it is necessary to display a window that lists all the post-translational modification islands ("PTM islands"). That window shows up when the user clicks onto the *View PTM islands* button shown in Figure 5.4, "The main window's Identifications tab".

5.3   Exploring PTM Islands Identification Data

The workflow involved in the scrutiny of phospho-proteomics data is comparable to that for a conventional proteomis project, as described in Section 4.1, "The Protein List Window". There are differences, however, both in terminology and in the kind of data presented to the user that require a detailed review, performed in

the following sections. The first step is to show the PTM islands, which is the step analogous to showing the identified proteins. The second step is to look into a given PTM island by scrutinizing the PTM peptides that make it; this step is analogous to showing the peptide list for a given protein.

5.3.1   THE PTM ISLANDS LIST WINDOW

The *PTM island list* window in FIGURE 5.5, "PTM ISLANDS LIST WINDOW" displays, in a table view, all the PTM islands that were identified (see SECTION 2.2.6.2, "PROTEIN IDENTIFICATION IN PHOSPHO-PROTEOMICS PROJECTS").
[3]

The *PTM island list* table view has many columns that characterize each PTM island, as described in detail in the text below.

FIGURE 5.5: PTM ISLANDS LIST WINDOW

In the table view of all the PTM islands, each row corresponds to an island. It must be noted that multiple rows may appear as identical islands. This is not exactly true because, while the PTM islands appear to be exactly the same, these have been identified in proteins that are listed in the proteins database under different accession numbers (see the *accession* column of the table view).

The columns in the table view hold post-translational modification-specific data described below.

- *Checked*: if checked, the corresponding PTM island will be taken into account;
- *PTM island ID*: unambiguous identifier for the PTM island. The nomenclature is important and follows a precise syntax according to this scheme:

  ptm<letter1><number><letter2><number><letter3><number>, with the following meaning:

---

3   The data presented in the examples below come from an experiment published in T. Y. Delormel *et al.* 2022. In vivo identification of putative CPK5 substrates in *Arabidopsis thaliana*. *Plant Science*. DOI: HTTPS://DOI.ORG/10.1016/J.PLANTSCI.2021.111121 ↗ .

- <letter1><number>: identifies the group of proteins that share PTM islands;

- <letter2><number>: identifies the PTM island in the above group;

- <letter3><number>: identifies the accession number of the protein in the group above.

- *accession*: the accession number of the protein in the protein database file;

- *description*: the description of the protein in the protein database file;

- *ptm positions*: the number of positions in the PTM island that bear a post-translational modification;

- *spectrum*: number of distinct spectra that allowed defining this PTM island;

- *PTM spectrum*: number of distinct spectra that allowed the identification of more than one searched post-translational modifications;

- *sequence*: number of unique peptide sequences found in this PTM island;

- *multi PTM*: number of spectra that allowed the identification of more than one post-translational modification site;

- *start*: the position of the PTM island on the protein sequence (the first residue of the PTM island);

- *length*: the number of residues that encompass the PTM island.

In the same way as for the *Protein list* window's table view of the identified proteins, the cells of the table are "active": clicking onto any cell triggers the opening of a window that provides details about the corresponding PTM island.

5.3.2   Delving Inside the PTM Island Identification Data

The protein identifications list table view, as pictured in Figure 5.5, "PTM islands list window" is actually an active matrix where the user can easily trigger the exposition of the data that yielded any PTM island identification element of the table. This is simply done by clicking onto any cell of the table at the row matching the PTM island for which scrutiny of the data is requested.

Depending on the column at which the mouse click happens, there might be two different windows showing up:

- Clicking onto a cell in either the *accession* or *description* column opens the *Protein details* window shown in Figure 5.6, "Protein details window".

FIGURE 5.6: **PROTEIN DETAILS WINDOW**

This window provides details about the protein identified as bearing one or more post-translation modification(s). The peptides that were matched as PSMs are highlighted in yellow. The *AT3G56150* provides a link to an external resource[4]. The other informational data bits are self-explanatory.

- Clicking onto a cell in the *PTM island ID* column opens the *PTM peptide list* window for that specific island. That window is displayed in both figures FIGURE 5.7, "PTM PEPTIDES LIST WINDOW (FIRST COLUMNS)" and FIGURE 5.8, "PTM PEPTIDES LIST WINDOW (LAST COLUMNS)" in the following section (SECTION 5.4, "THE PTM PEPTIDES LIST WINDOW").

5.4   THE PTM PEPTIDES LIST WINDOW

Each PTM island is essentially defined by a set of related peptides that sport one or more post-translational modifications. The exploration of the PTM peptides thus involves looking into the different peptides of a PTM island. A number of characteristics of these peptides are described in the following text.

---

4 Here, the page HTTPS://WWW.ARABIDOPSIS.ORG/SERVLETS/TAIROBJECT?TYPE=LOCUS&NAME=AT3G56150 ↗ opens in the browser.

Every PTM island has, associated to it, a list of post-translationally modified peptides. This figure illustrates the first columns of the table view that lists all the peptides making a PTM island. The contextual menu visible near peptide ID *pepb87b23* will be detailed later (Section 5.4.1.2, "Ambiguities on the post-translational modification sites").

FIGURE 5.7: PTM PEPTIDES LIST WINDOW (FIRST COLUMNS)

modifs	start	length	top Evalue	theoretical MH+	delta MH+	top hyperscore	top PTM positions	observed PTM positions
1Y:28.03 4S:79.97 6S:79.97 15K:28.03	35	16	2.7e-09	2060.77	0.000265847	36.5	4 6	4 6
1Y:28.03 4S:79.97 6S:79.97 15K:28.03	35	16	0.00036	2060.77	0.000404847	21.6	4 6	4 6
1Y:28.03 4S:79.97 6S:79.97 15K:28.03	35	16	1.4e-10	2060.77	0.00327185	30	4 6	4 6
1Y:28.03 4S:79.97 6S:79.97 15K:28.03	35	16	0.024	2060.77	0.00140985	17.6	4 6	4 6
1Y:28.03 4S:79.97 6S:79.97 15K:28.03	35	16	5.5e-11	2060.77	0.00157785	39.6	4 6	4 6
1Y:28.03 4S:79.97 6S:79.97 15K:28.03	35	16	0.00035	2060.77	0.00179385	24.4	4 6	4 6
1Y:28.03 4S:79.97 6S:79.97 15K:28.03	35	16	1.1e-10	2060.77	7.38466e-05	37.5	4 6	4 6
1Y:28.03 4S:79.97 6S:79.97 15K:28.03	35	16	0.0018	2060.77	0.00150185	17.9	4 6	4 6
1Y:28.03 4S:79.97 6S:79.97 15K:28.03	35	16	4e-09	2060.77	0.00209085	36.3	4 6	4 6
1Y:36.08 4S:79.97 15K:36.08	35	16	6.6e-07	1996.9	0.000649549	32.4	4	4
1Y:36.08 6S:79.97 15K:36.08	35	16	1.5e-05	1996.9	0.00314155	25.6	6	6
1Y:36.08 4S:79.97 15K:36.08	35	16	1.8e-07	1996.9	-0.00163045	36.6	4	4
1Y:32.06 4S:79.97 15K:32.06	35	16	3.5e-05	1988.86	0.00194488	32.9	4	4 6
1Y:32.06 4S:79.97 15K:32.06	35	16	2.2e-12	1988.86	0.00276688	47.2	4	4
1Y:32.06 4S:79.97 15K:32.06	35	16	3.9e-05	1988.86	0.00415188	24.3	4	4 6
1Y:28.03 4S:79.97 15K:28.03	35	16	4.2e-05	1980.81	0.000237847	26.8	4	4
1Y:28.03 6S:79.97 15K:28.03	35	16	4.2e-05	1980.81	0.000237847	26.8	6	6
1Y:28.03 4S:79.97 15K:28.03	35	16	1.5e-08	1980.81	0.00133285	36	4	4
1Y:28.03 6S:79.97 15K:28.03	35	16	1.5e-08	1980.81	0.00133285	36	6	6
1Y:28.03 4S:79.97 15K:28.03	35	16	3.8e-07	1980.81	0.000709847	36.3	4	4

Every PTM island has, associated to it, a list of post-translationally modified peptides. This figure illustrates the last columns of the table view that lists all the peptides making a PTM island.

FIGURE 5.8: PTM PEPTIDES LIST WINDOW (LAST COLUMNS)

The *PTM peptide list* window contains a large set of peptide characteristics organized in a number of columns, as described below:

- *peptide ID*: the unambiguous identity of the peptide;

- *sample*: the file name of the sample in which the peptide was found and sequenced;

- *scan*: the scan number of the MS/MS spectrum in which the precursor peptidic ion was fragmented;

- *RT*: the retention time at which the peptide eluted;

- *charge*: the charge of the peptidic ion;

- *sequence*: the sequence of the peptide. Note that the residues that do (or might) bear a post-translational modification are printed in red color;

- *modifs*: a semi-column-separated list of modified positions. The modification is identified by the net mass change that occurs upon the chemical modification. The *1Y:32.06 4S:79.97 15K:32.06* example indicates that the tyrosine at position 1 is dimethylated, the serine at position 4 is phosphorylated and that the lysine at position 15 is methylated (the dimethyl modification was commented above).

- *start*: the position of the peptide's first residue in the protein sequence;

- *length*: the number of residues in the peptide;

- *top Evalue*: best Evalue from those calculated for different peptide spectrum matches that occurred in a single fragmentation scan (see Section 5.4.1.2, "Ambiguities on the post-translational modification sites" below for a thorough description of this situation);

- *theoretical MH+*: the calculated mass of the $[M+H]^+$ peptidic ion;

- *delta MH+*: the difference between the measured and the calculated $[M+H]^+$ masses;

- *top hyperscore*: best Hyperscore from those calculated for different peptide spectrum matches that occurred in a single fragmentation scan (see Section 5.4.1.2, "Ambiguities on the post-translational modification sites" below for a thorough description of this situation);

- *top PTM positions*: positions of modified residues for the peptide having the best Evalue (see above) for the current scan;

- *observed PTM positions*: space-separated list of all the modified positions found for the current scan.

5.4.1 Delving Inside the PTM Peptide Identification Data

The *PTM peptide list* table view (Figure 5.7, "PTM peptides list window (first columns)") is actually an active matrix where the user can easily trigger the exposition o fthe data that yielded any PTM peptide identification element of the table. This is simply done by clicking onto any cell of the table at the row matching the peptide for which scrutiny of the data is requested.

5.4.1.1 The Peptide Details Window

Clicking a cell in the *peptide ID* column opens the *Peptide details* window shown in Figure 5.9, "Peptide details window". Notice how this window is similar to the one described for conventional non PTM-based projects at Section 4.2.3.1, "The Peptide Details Window".

This window displays a large amount of informational data bits that characterize the MS/MS spectrum *vs* peptide match (PSM) for the peptide ID that was clicked in the *PTM peptide list* table view window. Almost all the information data bits shown in this figure are self-explanatory.

FIGURE 5.9: PEPTIDE DETAILS WINDOW

TIP

The nomenclature of the product ions in the MS/MS spectrum shown in the figure above is simple: when the ion under a given MS/MS spectrum peak is the result of a neutral loss of phosphoric acid, the ion is labelled "y*x*(-P*z*)", with *x* being the index of the y ion, and (-P*z*) indicating the loss of *z* phosphoric acid neutral molecule(s).

When ions actually bear any number of post-translational modifications, these are not listed along with the ion series (*b* or *y*) and index text because that would quickly become unwieldy, from a graphical point of view.

5.4.1.2 AMBIGUITIES ON THE POST-TRANSLATIONAL MODIFICATION SITES

The *PTM peptide list* table view in FIGURE 5.7, "PTM PEPTIDES LIST WINDOW (FIRST COLUMNS)" lists the *pep-b87b23* peptide ID thrice because that peptide was identified in three different MS/MS scans (see the *scan* column values for the three rows). What is interesting with these three rows, is that the peptide sequence showing the modification position(s) varies from a row to the other. In one row, the sequence only shows a single red-colored serine residue. That serine was positively identified as a phosphorylated residue. In the other two rows,

one serine is red-colored and the other is orange-colored. For these two scans, the position of the phosphorylated residue is not ascertained: the MS/MS data only tell that it is possible that one or the other serine residue be phosphorylated (but not both).

 NOTE

The coloring of these two serine residues is arbitrary in this case: since the PSMs that yielded this same sequence are absolutely identical from a score point of view (when opening the peptide details window, one can check that the Evalues are identical for the two PSMs), the software labels in red the first modification position and in orange all the remaining ones.

When clicking any cell of any one of the two rows where there is an ambiguity over the location of the phosphorylation event, *i2MassChroQ* shows a contextual menu that displays the possible phosphorylation positions. The user can thus select one or the other of the menu items to display the details of the corresponding peptide.

The ambiguity about the phosphorylation site above, on *Peptide ID pepb87b23*, is interesting. Indeed, when the user selects the first item of the contextual menu and then (by keeping the `Ctrl` key pressed) selects the second item of that menu, the windows that open up show exactly the same informational data bits about the MS/MS scan: the two PSMs were calculated by the search engine on the basis of the very same MS/MS scan (FIGURE 5.10, "TWO MASS SPECTRUM VS PEPTIDE MATCHES (PSM) IN A SINGLE MS/MS SCAN"). This is one reason why both positions could not be ascertained: the engine says that one position is plausible (with a pretty low Evalue) and that the other position is equally plausible (with the exact same low Evalue).
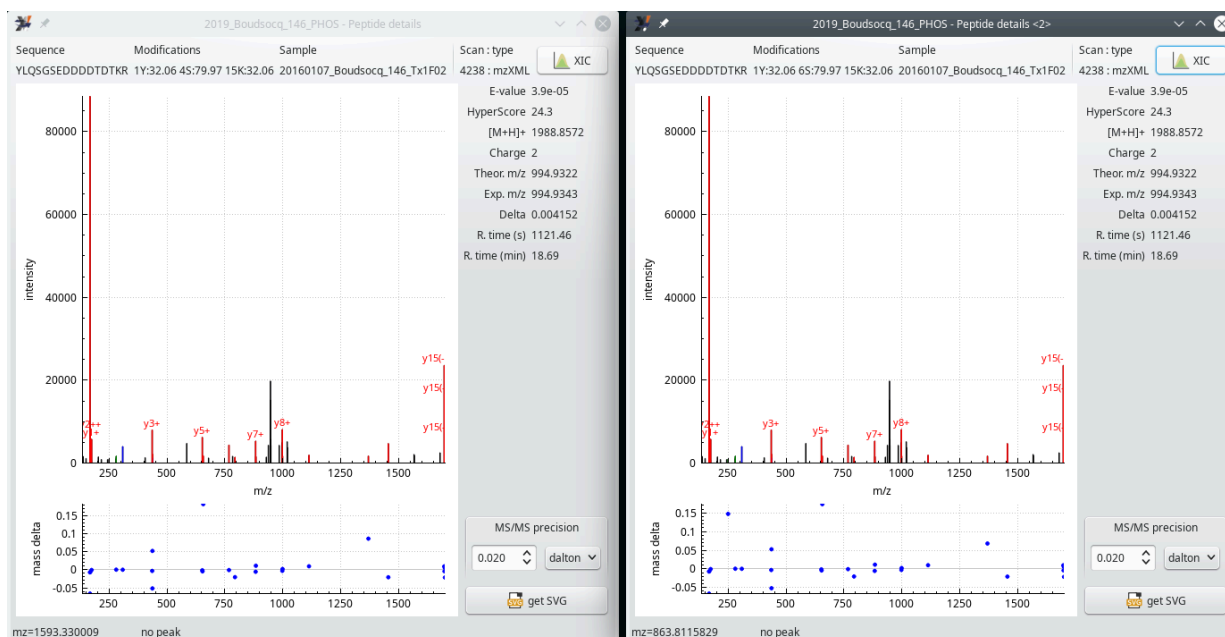
This figure shows the two *Peptide details* windows that are opened when the user clicks onto the first and the the second menu items of the contextual menu shown in the *PTM peptide list* window shown in FIGURE 5.7, "PTM PEPTIDES LIST WINDOW (FIRST COLUMNS)". In both windows, the scan number is the same (*4238*), demonstrating that both PSMs were computed from the same MS/MS spectrum acquisition. Further, the same pretty low E-value should hint at a low reliability of the phosphorylation site identification.

FIGURE 5.10: TWO MASS SPECTRUM *VS* PEPTIDE MATCHES (PSM) IN A SINGLE MS/MS SCAN

5.4.1.3 THE XIC VIEWER WINDOW FOR THE PTM PEPTIDE DETAILS

One interesting feature of the *Peptide details* window, is the *XIC* button (top right) that triggers the calculation of an extracted ion current chromatogram, as pictured in FIGURE 5.11, "THE EXTRACTED ION CURRENT (XIC) CHROMATOGRAM VIEWER WINDOW". Although very similar to the window described at SECTION 4.2.3.2, "THE XIC VIEWER WINDOW FOR THE PEPTIDE DETAILS", this phospho-proteomics-specific version of the *XIC viewer* has specific informational data bits described below.

The extracted ion current (XIC) chromatogram viewer shows the peptide sequence interspersed with post-translational modifications data. The modifications are listed as PSIMOD OBO accession number *(MOD:xxxxx)* text elements[5].

FIGURE 5.11: THE EXTRACTED ION CURRENT (XIC) CHROMATOGRAM VIEWER WINDOW

The phospho-proteomics-specific informational data bits provided in the *XIC viewer* window (see Figure 5.11, "The extracted ion current (XIC) chromatogram viewer window") are located right below the top border of the plot frame.

---

5 Montecchi-Palazzi *et al.* 2008. PSI-MOD community standard for representation of protein modification data. *Nature biotechnol.*

6 Advanced Proteomics Configurations

This chapter describes in detail all the various advanced configurations that determine the workings of *i2Mass-ChroQ* in various contexts.

Some of the advanced proteomics configurations can only be carried over when protein identification results have been loaded in *i2MassChroQ*. Indeed, only at that point, does the menu of the main program window show the submenus of interest, as shown in Figure 6.1, "The Edit menu showing various configuration tasks".
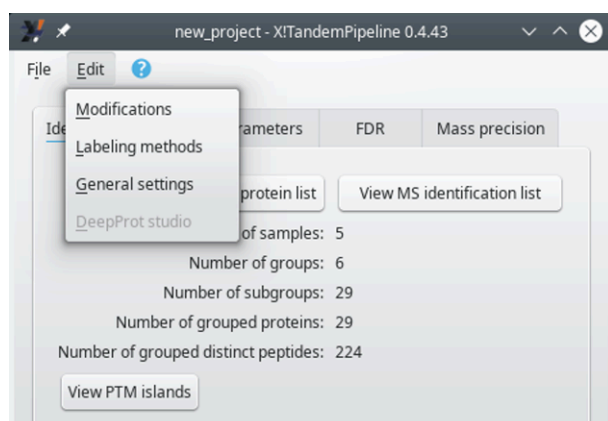

This menu becomes available in the main program window only when identification data have been loaded in *i2MassChroQ*

FIGURE 6.1: THE *EDIT* MENU SHOWING VARIOUS CONFIGURATION TASKS

6.1 Configuring Modifications

It is often required to specify the chemistry of a given chemical modification that might be performed (or spontaneously observed) on the proteins of a sample. For example, proteins in a sample often undergo a chemical modification of the cysteine residues in their reduced form to block them from reoxidizing into a sulphide bond. The reagent used is iodoacetamide and the reaction name is *carbamidomethylation*.

In order to be able to access the modifications configuration, select *Modifications* from the *Menu* menu. The window that opens up is shown in Figure 6.2, "Modifications editor window (folded)". In that *Modifications editor* window, the currently defined modifications are listed with no details about them.
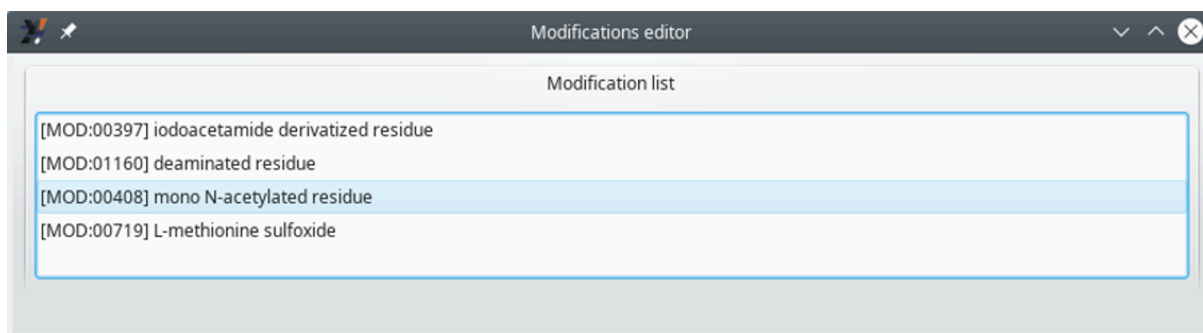
When the *Modifications editor* window is displayed, all the modifications data are folded.

FIGURE 6.2: MODIFICATIONS EDITOR WINDOW (FOLDED)

To start editing an existing modification, click the correponding item from the list, which opens up a *Selected modification* group box widget, as shown in FIGURE 6.3, "MODIFICATIONS EDITOR WINDOW (UNFOLDED)".
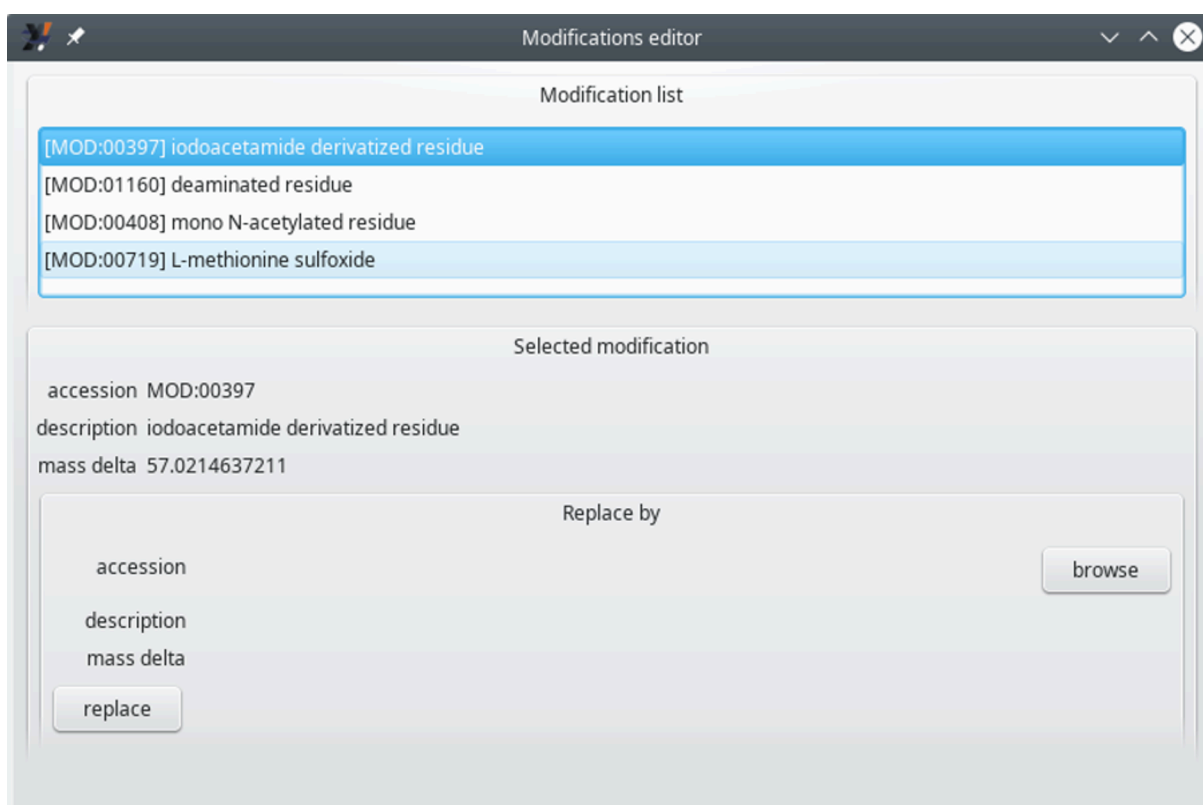
This view shows the details of the modification item currently selected.

FIGURE 6.3: MODIFICATIONS EDITOR WINDOW (UNFOLDED)

The *Modifications editor* window displays the details of the modification currently selected in the *Modifications list*. The main informational data bits are the following:

- *accession*: the PSI-MOD[1] identificator. This value matches the id field of the corresponding PSI-MOD ontology record (see an example below);

- *description*: the description of the PSI-MOD entity. This value corresponds to the name field of the corresponding PSI-MOD ontology record;

- *mass delta*: the net mass change that results from a protein modification using this modification entity.

NOTE: A TYPICAL TERM DEFINITION FROM THE PSI-MOD ONTOLOGY

```
[Term]
id: MOD:00397
name: iodoacetamide derivatized residue
def: "A protein modification that is produced by reaction
with iodoacetamide, usually replacement of a reactive hydrogen
with a methylcarboxamido group."
[PubMed:11327326, PubMed:11510821, PubMed:12422359, Unimod:4]
subset: PSI-MOD-slim
synonym: "Carbamidomethyl" RELATED PSI-MS-label []
synonym: "Iodoacetamide derivative" RELATED Unimod-description []
xref: DiffAvg: "57.05"
xref: DiffFormula: "C 2 H 3 N 1 O 1"
xref: DiffMono: "57.021464"
xref: Formula: "none"
xref: MassAvg: "none"
xref: MassMono: "none"
xref: Origin: "X"
xref: Source: "artifact"
xref: TermSpec: "none"
xref: Unimod: "Unimod:4"
is_a: MOD:00848 ! reagent derivatized residue
```

If a change, that is, a replacement, is required in the modification, the first step is to click onto the *browse* button located in the *Replace by* group box, which opens the *PSI MOD selection* window shown in FIGURE 6.4, "PSI MOD SELECTION WINDOW".

---

1 The PSI-MOD ontology is perusable at HTTPS://RAW.GITHUBUSERCONTENT.COM/HUPO-PSI/PSI-MOD-CV/MASTER/PSI-MOD.OBO ↗
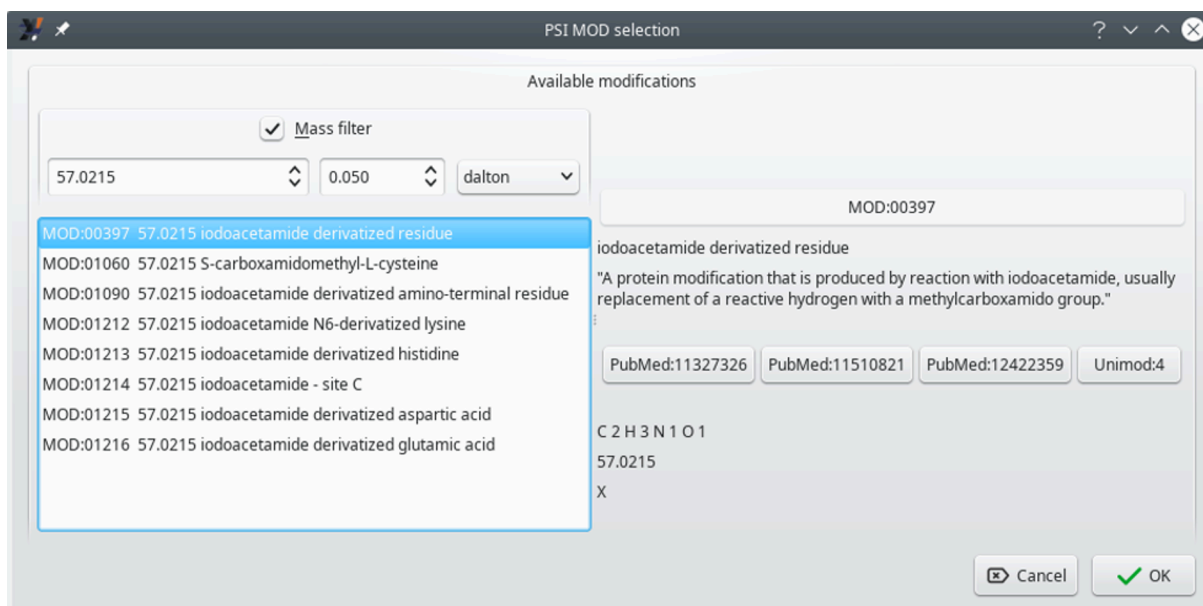
This window allows one to browse the PSI-MOD ontology on the basis of the modification net mass value.

FIGURE 6.4: PSI MOD SELECTION WINDOW

The window shows the various PSI-MOD-defined ontology terms that match the net mass value of the modification that was selected when the *browse* button was clicked.

It is possible to search for other PSI-MOD terms that match a different net mass change by modifying the mass value in the spin box widget that is found under the control of the *Mass filter* check box. The other spin box widget next to the first one allows one to enter the tolerance of the net mass change match. Available "units" for that tolerance are Dalton, ppm and res.

If the *Mass filter* check box is unchecked, the list beneath it displays *all* the PSI-MOD ontology terms.

Once the desired term has been found and selected, click the *OK* button. In order to effectively replace the old term with the new one, do not forget to click the *replace* button.

⊙ Tip: Why bothering with PSI-MOD ontology modification terms ?

It is beneficial to define chemical modifications based on the PSI-MOD ontology terms because the defined modification is then used as its code (for example *MOD:00397*) in the *X!Tandem* presets chemical modifications sections (see Section 3.3, "Setting the X!Tandem Run Presets"). Indeed, instead of only entering the net mass, defining a modification preset with the PSI-MOD ontology term code allows *i2MassChroQ* to compute a number of informative supplementary data. For example, when an isotopic cluster is simulated for a peptide that is modified with a PSI-MOD-documented modification, the formula of the modification recorded in the term is taken into account in the isotopic cluster calculation.

6.2 Configuring Labeling Methods

i2MassChroQ can handle stable isotope labeling methods. In order to list the available methods, select the *Label-ing methods* menu item of the *Edit* menu in the main program window. The *Labeling methods editor* window that opens up is shown in Figure 6.5, "Labeling methods editor window".
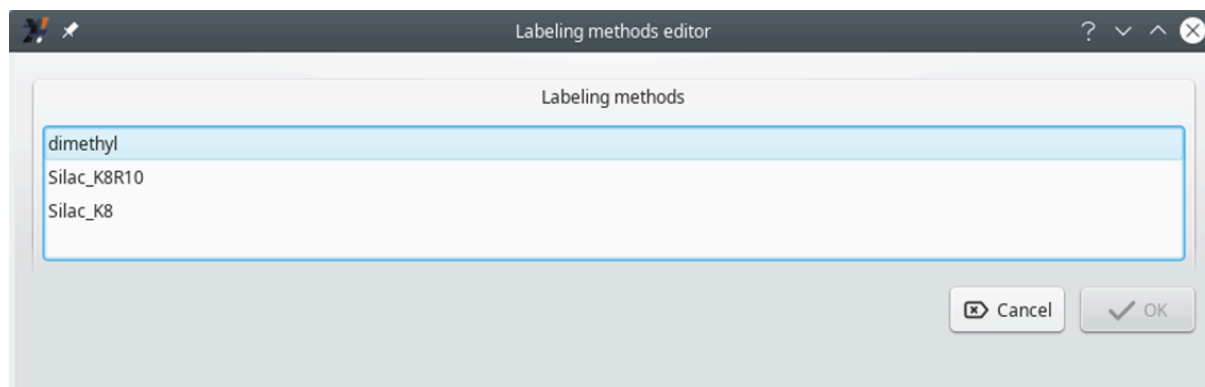
This window lists all the available labeling methods.

FIGURE 6.5: LABELING METHODS EDITOR WINDOW

Configuring a labeling method is as easy as selecting one item from the list shown in Figure 6.5, "Labeling methods editor window" and clicking *OK*. If unsure about the selection, just click *Cancel*.

## ✋ Warning: It is not possible to backstep

Once the user has clicked the *OK* button, it is no more possible to change the labeling method without recomputing the whole data set.

Using a labeling method involves a crucial step during the *X!Tandem* configuration, as described in Section 3.3, "Setting the X!Tandem Run Presets" and in Caution: Phospho-proteomics projects often involve la-bel-based quantification. Whenever an experiment involves chemical labeling of any protein residue, the sci-entist is expected to configure the labeling modifications in the *X!Tandem* presets before running it to search the database, so that the engine models correctly-modified peptides.

When the *X!Tandem* database search engine has finished identifying peptides and proteins, the modified pep-tides are only tagged using the modifications' net masses. This is visible in the peptide details window shown in the upper left part of Figure 5.9, "Peptide details window". There, one can see that the peptide is modified on its lysine 15 residue with one of the labels defined in the *Fixed modifications 1* field of the presets configuration window shown in Figure 5.3, "Configuring the phosphorylated residues".

Selecting a labeling method, as described above, will allow *i2MassChroQ* to report residue modifications in a much more detailed and informative way. For example, the name of the chemical modification will be used in-stead of the net mass change due to that modification. For this to happen and the effects to be visible, *i2Mass-*

ChroQ goes back through all the identification process and makes sure that the modified peptides are tagged according to the modifications specified in the labeling method. The effects are visible in the *XIC viewer* window as shown in Figure 6.6, "Chemical modifications are tagged with their PSI-MOD code ".
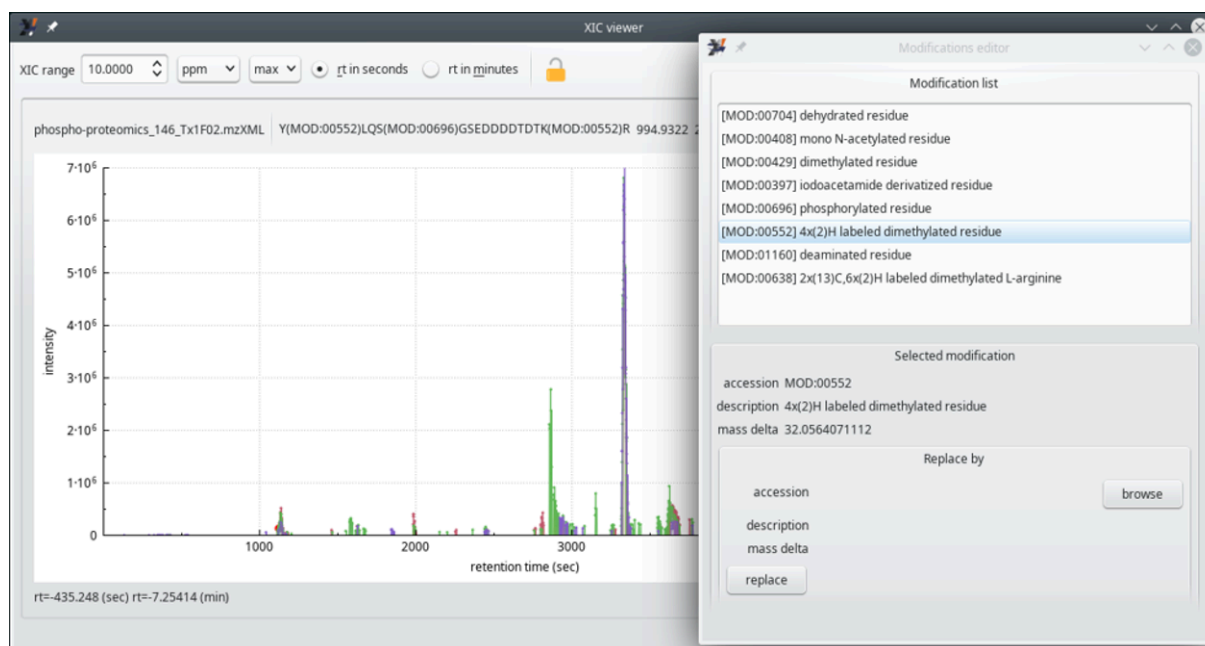

When the user selects a labeling method from the list of available chemical modifications, the chemically modified residue gets an associated tag with the PSI-MOD code. This is visible in this figure, at the top line of the *XIC viewer* window, where the last peptide's arginine residue is modified with *MOD:00552*, that corresponds to the *4x(2)H labeled dimethylated residue* modification referenced in the *Modification editor* window on the right hand side of this figure.

FIGURE 6.6: CHEMICAL MODIFICATIONS ARE TAGGED WITH THEIR PSI-MOD CODE

6.3 The *i2MassChroQ* General Settings

There are general settings that might be configured for the whole *i2MassChroQ* program. These settings are available upon selecting the *General settings* menu of the *Edit* menu of the main program window. The window that opens up is shown in Figure 6.7, "General settings window".
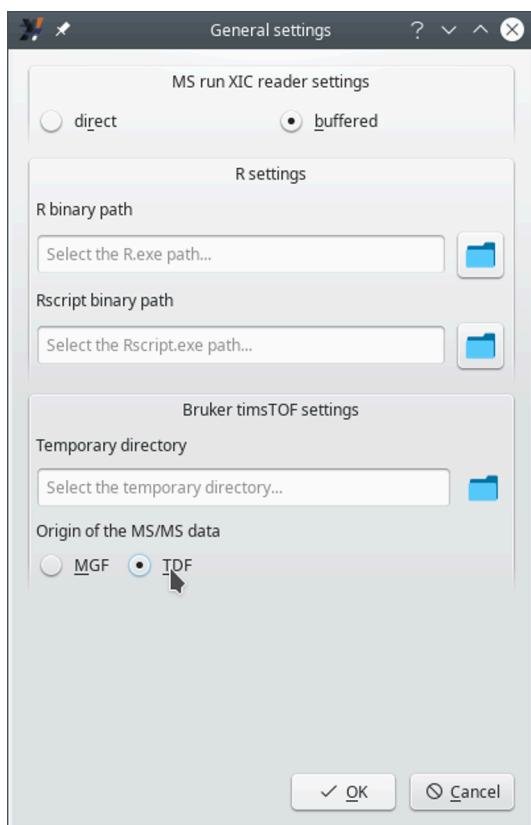
General settings might be configured in this window, as described below.

FIGURE 6.7: **GENERAL SETTINGS WINDOW**

The settings that can be configured are detailed below.

- *MS run XIC reader settings*: if *direct* is selected, the XIC chromatogram is computed by read the MS data right from the file; if *buffered* is selected, the XIC chromatogram is computed by using a cache mechanism that buffers data in the computer's memory.

- *R settings*:

    - *R binary path*: full path to the R program;

    - *Rscript binary path*: full path to the Rscript program;

- *Bruker timsTOF settings*: the fields below need filling-in if the data in the proteomics project originated in the Bruker's timsTOF line of instruments.

    - *Temporary directory*: optionally set the full path to the directory where the program stores temporary data;

    - *Origin of the MS/MS data*: if *MGF* is selected, the data are read from the Bruker-generated MGF file; if *TDF* is selected, the data are read from the Bruker native TDF file.

✋ Warning

If the user selects the *MGF* option above (because this is what they have), *i2MassChroQ* will not be able to extract XIC chromatograms, because the MGF file format does not contain MS data, it contains only MS/MS data. If having the native TDF files at hand (in the .d directory), the user is advised to select *TDF*.

7 *i2MassChroQ* and Quantitative Proteomics

This chapter describes in detail the way to prepare the work that will be carried over by the *MassChroQ* module.

7.1 Interface to the *MassChroQ* Quantitative Proteomics Module

While it is certainly possible to perform pretty thorough analyses by exploring data by way of peptide identification—protein inference scrutiny strategies, it is necessary to expand the boundaries of these strategies if quantitative proteomics projects are being developed. We have now integrated *MassChroQ* in *i2MassChroQ*, which makes it straightforward to perform quantitative proteomics work right after the identification—protein inference process.

The way the *MassChroQ* program is harnessed in *i2MassChroQ* is according to the following outline:

- Open an *i2MassChroQ* project or load protein identification results files;

- Configure all the aspects of the *MassChroQ* run in a specific *MassChroQ* configuration window;

- Use *i2MassChroQ* to run the external *MassChroQ* software or have *i2MassChroQ* only write the file that *MassChroQ* uses to perform its quantitative proteomics task at a later stage and outside of *i2MassChroQ*.

7.1.1 Preparing sample associations for *MassChroQ*

Performing quantitative proteomics experiments most likely involves comparing samples between them. That means that most often multiple samples need to be associated into meaningful groups. Before going on with the *MassChroQ* configuration, it is thus necessary to first define the sample associations. In fact, since a given sample is actually a given LC-MS run, and that each MS run's data are then used to perform protein identifications, these assocations are performed between MS runs.

To perform a quantitative proteomics experiment, the very first step is to load either the protein identification results (see Section 3.4, "Loading the Protein Identification Results" or an <u>xpip</u> project file (see Section 3.5, "Loading i2MassChroQ projects").

Once the protein identification results have been loaded (or the *i2MassChroQ* project file), the sample associations (that is, between MS run files) need to be performed by first clicking onto the *View MS identification list* button of the main program window (see Section 3.4.2, "Displaying the MS Identifications List"). The MS runs are displayed in a table and sample associations can be performed by right-clicking onto the cells of the *Alignment group* column label, as shown in Figure 7.1, "Defining sample associations for XIC alignments".

NOTE

The sample (MS run) associations are critical not only because one wants to compare quantitative data about somehow related samples, but also because of the way *MassChroQ* performs quantification of proteomics data. Indeed, *MassChroQ* uses not *spectral count*-based strategies but an *area under the curve* strategy where the area of mass peaks is determined by looking at XIC chromatograms for these mass peaks. The associations will thus allow the software to perform the alignment of the XIC chromatograms that will be essential for the quantification analysis. Indeed, even LC-MS runs of an identical sample will not provide identical (m/z,retention time) pairs. But, to be able to quantify proteomics data on the basis of the area under the curve of XIC chromatogram peaks, it is necessary that all the XIC chromatograms for all the associated samples be properly aligned.

The associations between samples can be performed in any arbitrary way, according to the user's experimental scheme. Any number of groups can be defined that may contain any number of samples. The process is described in FIGURE 7.1, "DEFINING SAMPLE ASSOCIATIONS FOR XIC ALIGNMENTS" and FIGURE 7.2, "SAMPLE ASSOCIATIONS ARE DONE BY GROUPING SAMPLES INTO GROUPS".
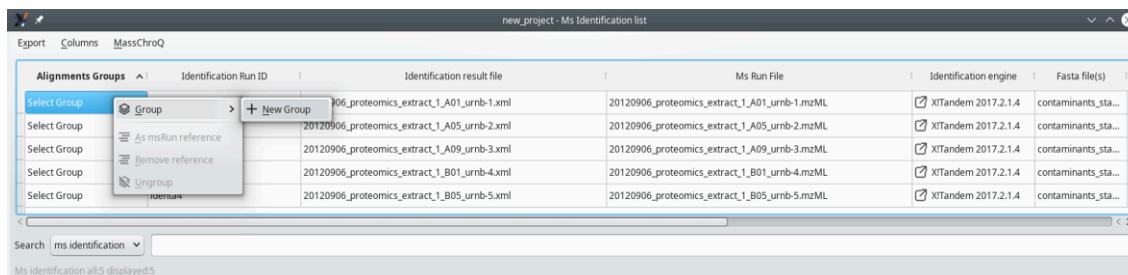
By right-clicking into the cells of the *Alignment groups* column, groups can be defined and samples can be associated to the groups.

FIGURE 7.1: **DEFINING SAMPLE ASSOCIATIONS FOR XIC ALIGNMENTS**

Three groups have been defined, two groups having two samples each and one group having only one sample.

FIGURE 7.2: SAMPLE ASSOCIATIONS ARE DONE BY GROUPING SAMPLES INTO GROUPS

Tip: Sample associations with specific sample sets

Sample associations play a critical role when samples (that is, MS runs) have conceptual relationships. For example, let's assume that a project used polyacrylamide gel electrophoresis as a protein separation method. Five related samples (be them biologically-relevant variants or technical replicates, for example) have been loaded onto five different lanes of the gel. The migration pattern between the five lanes is very similar and one could observe reproducible bands (albeit with different intensities) from one lane to the other, say, in sample 1 a band A, below a band B and so on. Sample 2 would also have that pattern, with a band A and a band B, and the same for the remaining samples (that is, lanes). Bands would be excised and subjected to trypsin digestion, the peptides would be extracted and analysed by mass spectrometry. The sample associations, here, would typically involve the definition of groups that associate related "horizontal" bands on the gel. For example, group A would associate all the bands A from the five samples, group B would associate all the bands B from the samples and so on. The sample associations would thus allow the quantification and comparison of kin proteins from the various samples.

The alignement of XIC chromatograms computed for samples from a given association group is performed by having one reference sample in that group. Each group must have a reference sample. The definition of the reference sample can be performed by the user at this stage (or at a later stage, described later) by using the context menu shown in Figure 7.3, "Setting the reference sample for the alignment".
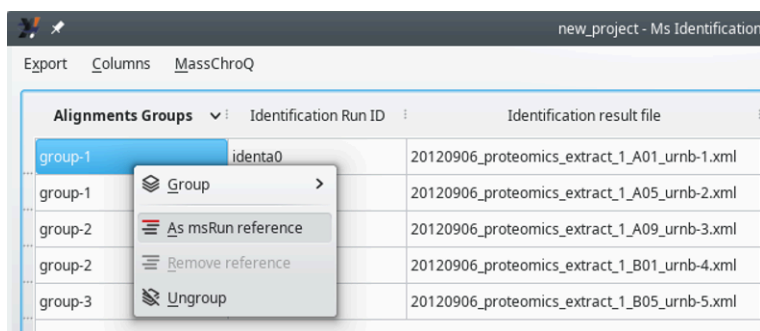
Use the context menu by right-clicking on the cells in the *Alignment groups* column to set the alignment reference in each group.

FIGURE 7.3: **SETTING THE REFERENCE SAMPLE FOR THE ALIGNMENT**

◈ NOTE

Selecting the proper alignment reference is not something to do without thinking because the reference sample will serve as the basis for the alignement of all the samples in the group. The best sample to be chosen as alignement reference is the sample that shares the most precursor ions' m/z values with all the other samples. It is possible to delegate to *i2MassChroQ* the choice of the alignment reference sample, as described later.

Now that the sample associations have been performed, the next step is to configure *MassChroQ* from within *i2MassChroQ*. This is described in the next sections.

7.1.2 CONFIGURATION OF *MassChroQ*

i2MassChroQ provides an interface to *MassChroQ*, the software that performs XIC extractions for a list of precursor ions' m/z values. That interface is shown by selecting the *MassChroQ* menu item of the main *File* menu. The window that opens up is shown in FIGURE 7.4, "THE MASSCHROQ INTERFACE WINDOW (SAMPLE ASSOCIATIONS)", and is described below.
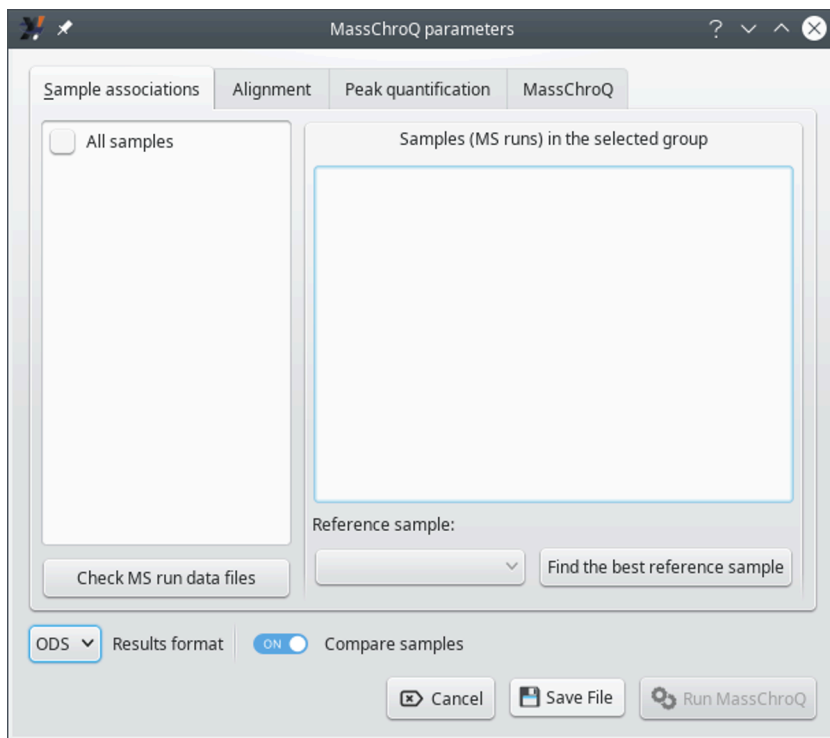
This window offers an interface to the *MassChroQ* program. The *Sample associations* tab allows one to define groups of samples that will be processed together. All the configurations in the tabs are described in the sections below.

FIGURE 7.4: THE *MASSCHROQ* INTERFACE WINDOW (SAMPLE ASSOCIATIONS)
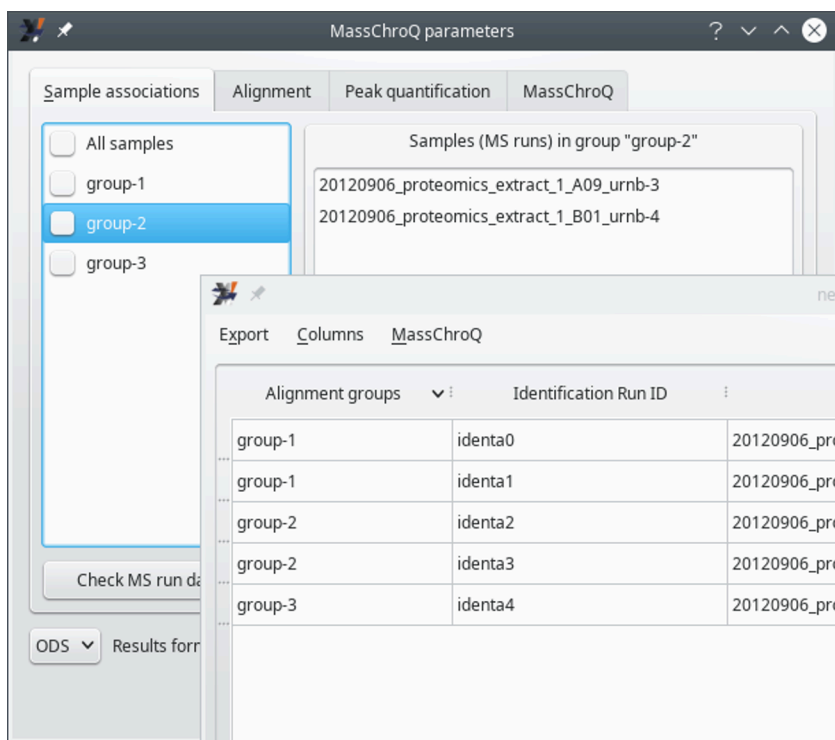
7.1.2.1   THE *Sample associations* TAB

This tab allows one to configure the sample associations. The window state shown in FIGURE 7.4, "THE MASS-CHROQ INTERFACE WINDOW (SAMPLE ASSOCIATIONS)" corresponds to a situation in which the user did not define sample associations according to the way described in SECTION 7.1.1, "PREPARING SAMPLE ASSOCIATIONS FOR MASSCHROQ". In this case, it is assumed that the user wants to treat all the samples as a single group, (the *All_samples* group). To reveal all the samples (that is, MS runs) that are being handled, check the *All_samples* check button, which will associate all the samples in that single group and display them in the right hand side list widget, as shown in FIGURE 7.5, "THE MASSCHROQ INTERFACE WINDOW (SAMPLE ASSOCIATIONS) - ALL SAMPLES LISTES".
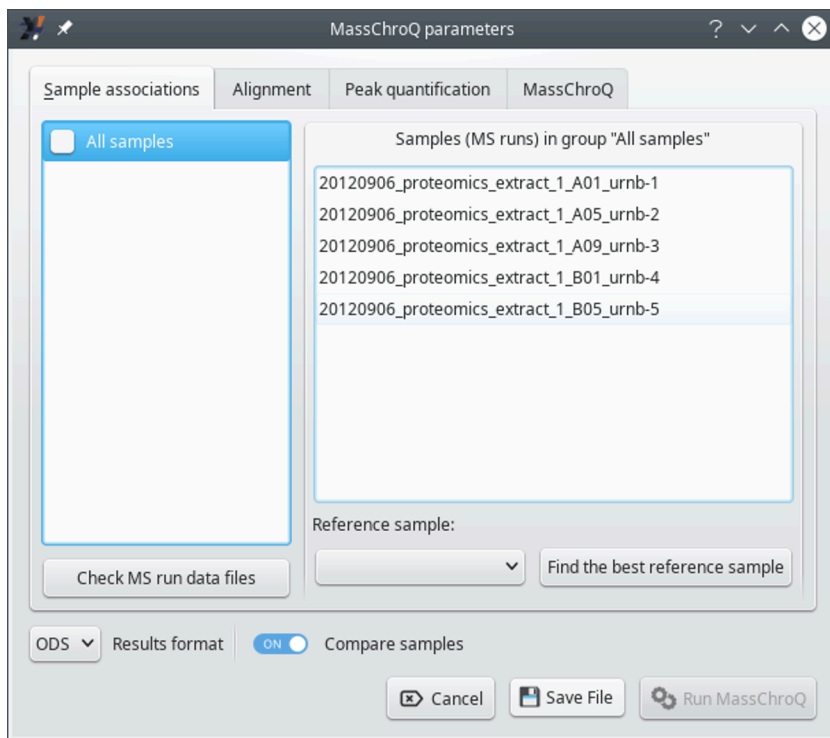
By checking the *All_samples* check box on the left hand side list widget, all the samples in the project are associated in a single *All_samples* group and displayed in the list widget on the right hand side of the window.

FIGURE 7.5: THE *MASSCHROQ* INTERFACE WINDOW (SAMPLE ASSOCIATIONS) - ALL SAMPLES LISTES

If the user has crafted groups of associated samples, as described in Section 7.1.1, "Preparing sample associations for MassChroQ", the window displays different settings at start (see Figure 7.6, "The MassChroQ interface window (Sample associations) - pre-defined sample associations ").

When the sample associations were defined before opening the *MassChroQ* interface window (the inserted window corresponds to SECTION 7.1.1, "PREPARING SAMPLE ASSOCIATIONS FOR MASSCHROQ"), the groups of associated samples are displayed in the list widget on the left hand side of the window. Selecting group names in that list allows one to display the samples associated in a given group. To include a group in the *MassChroQ* computations, check the corresponding check box widget.

FIGURE 7.6: THE *MASSCHROQ* INTERFACE WINDOW (SAMPLE ASSOCIATIONS) - PRE-DEFINED SAMPLE ASSOCIATIONS

To verify which samples are being associated in a given group, select that group in the list widget on the left hand side of the window.

To make sure a given group is going to be accounted for by *i2MassChroQ* during the preparation of the file that lists all the precursor ions' peaks for which the XIC extractions needs to be performed at a later stage by *MassChroQ*, check the corresponding check box.

The *Check MS run data files* button allows the user to make sure that all the samples associated in the various groups can be found as mass spectrometry data files (`mzML` or `mzXML` files). This is a hard requirement because *MassChroQ* does the quantification of peptide mass spectrometric signals by extracting ion current for the peptide's precursor ion (XIC extraction). For this to be possible, the software needs to access the mass spectrometry data files.

The *Reference sample* drop-down list widget allows one to select the alignment reference sample for the currently selected sample association group in the left hand side list. The alignment reference sample must be chosen with care, as explained in FIGURE 7.3, "SETTING THE REFERENCE SAMPLE FOR THE ALIGNMENT".

TIP

If the selection of an alignment reference sample is not possible, the user might ask *i2MassChroQ* to search for it by clicking the *Find the best reference sample* button. *i2MassChroQ* will look into all the sample files associated in the current group and search for the sample that shares the maximum number of precursor ions with all the other samples. The discovered MS run file is then set to the drop-down list widget.

The *Results format* drop-down list widget allows the user to select the kind of format that the quantification results should be written in. The *ODS* format is the standard format for the *LibreOffice* software suite. The *TSV* format is a "tab-separated values" text format.

The *Compare samples* switch indicates if the results output file should display a low-details version of the data but arranged in a manner that allows the user to easily compare the quantification data about the various samples.

7.1.2.2 THE *Alignment* TAB

This tab allows one to configure the way the XIC chromatograms obtained for the different associated samples are aligned (see SECTION 7.1.1, "PREPARING SAMPLE ASSOCIATIONS FOR MASSCHROQ") as shown in FIGURE 7.7, "THE MASSCHROQ INTERFACE WINDOW (ALIGNMENT)".
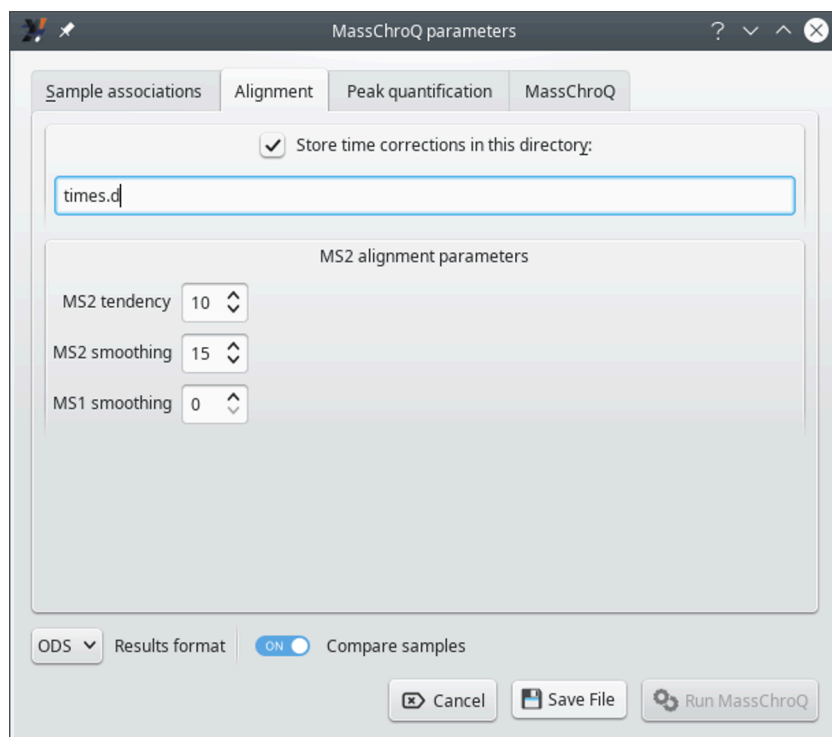
This tab configures the way *i2MassChroQ* performs the XIC chromatograms alignment between associated samples in the various groups. If the user is interested in the results of the alignment, the XIC retention time corrections can be stored in the directory specified at *Store time corrections in this directory* for later scrutiny.

FIGURE 7.7: THE *MASSCHROQ* INTERFACE WINDOW (ALIGNMENT)

The *MS2 alignment parameters* group box widget gathers parameters that are critical to the XIC chromatogram alignment algorithm for all the samples associated in a given group, as described below.

- *MS2 tendency*: half size of the window used to apply a moving median on the MS/MS retention time deviation curve. Used to create the tendency deviation curve. Of course the appropriate value for this window depends on the number of identified peptides that the two runs (reference run and run being aligned) have in common. Usually a good value is 10. While aligning, *MassChroQ* outputs the number of peptides in common which can be used to readjust this parameter if necessary.

- *MS smoothing*: half size of the window used to apply a moving average on the MS/MS retention time deviation curve. Smooths the deviation curve. Same as the above parameter, usually a good value is 10.

- *MS1 smoothing*: half size of the window used to apply a moving median on the MS retention time corrections curve. This smoothing parameter is optional, and it is not necessary most of the time. It could be used in place of the MS2 smoothing parameter in cases of a small number of shared identified peptides (< 100), in which case a good value is 20.

7.1.2.3 The *Peak quantification* Tab

This tab allows one to configure the way the peaks in the XIC chromatograms are evaluated from a quantification stand point. These parameters need some testing as they might depend on the instrument whence the data originated.
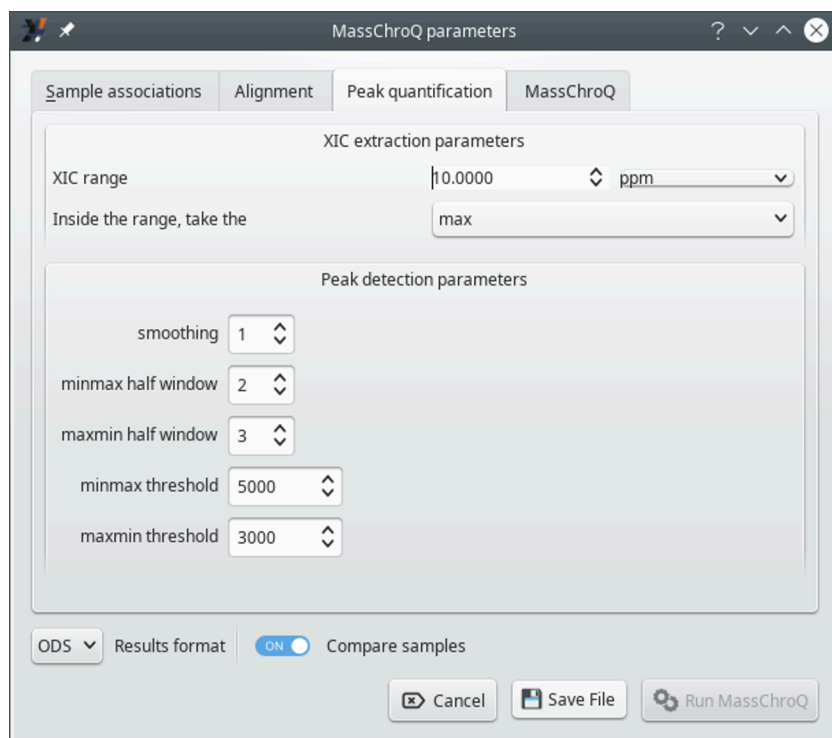
This tab configures the way *i2MassChroQ* performs the evaluation of the peaks in the XIC chromatograms from a quantification stand point. The settings in this dialog window might need some tweaking as they might depend on the instrument whence the data originated.

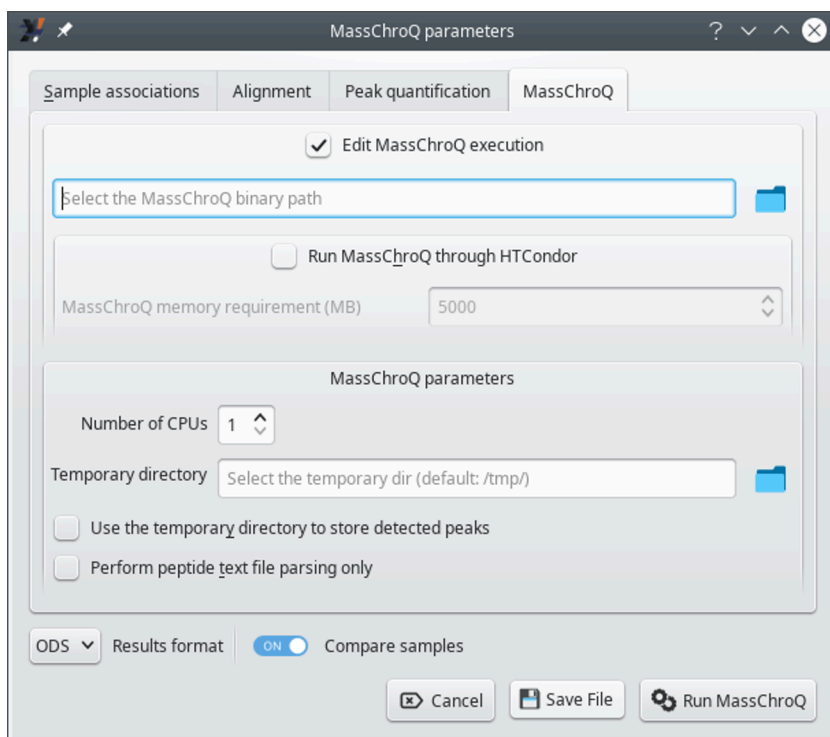FIGURE 7.8: **THE *MASSCHROQ* INTERFACE WINDOW (PEAK QUANTIFICATION)**

- *XIC extraction parameters*: these parameters govern the way the program searches for m/z values in the mass spectral data.

  - *XIC range*: the m/z width (mass tolerance) for searching m/z values in the mass data during the XIC extraction. Units can be part-per-million (*ppm*), resolution (*res*) or Dalton (*dalton*). The wider the window, the rougher the XIC extraction. This value typically depends on the resolving power of the instrument that acquired the data.

  - *Inside the range, take the*: once the m/z window has been located in the mass spectral data, it will contain a nubmer of points. This settings determines what kind of signal intensity to compute for the m/z window (that is, what to do with the m/z points contained in the m/z window). If *max* is

selected, only the max-intensity point in the m/z window is used as the signal intensity corresponding to the m/z window. If *sum* is selected, the sum of the intensities of all the m/z points in the window is used.

- *Peak detection parameters*: these parameters govers the way the program detects peaks.

  - *smoothing*: number of points around the point being considered in the XIC chromatogram. If set to one, the rolling window will contain three points: one before the considered point, one after it and the considered point itself. This setting thus determines the width of the rolling window that is used to iterate in the XIC chromatogram in search for peaks. This window, whatever the setting, will shift by one point at each iteration in the XIC chromatogram.

  - *minmax half window*: the half window size used to apply the close (min/max) transform on the XIC intensities. This window determines the number of scan points over which two peaks will be considered separately, otherwise they would have been merged. A good half window value is usually 3 (which makes a window of 7).

  - *maxmin half window*: same as above but for the close (max/min) transform. This window determines the minimum peak width (in scan points number) below which the peak would not be detected. A good half window value is usually 2 (which makes a window of 5).

  - *minmax threshold*: threshold on the close signal: a minimum intensity value below which peaks are not detected on the closing signal. This threshold is usually two or three times the background noise intensity level, which depends on your mass spectrometer.

  - *maxmin threshold*: threshold on the open signal: a minimum intensity value below which peaks are not detected. It corresponds to the opening signal upper limit and it represents the background signal upper level. A good value would thus be slightly bigger than your background noise intensity level.

The *MassChroQ* Tab

This tab allows one to configure the way *MassChroQ* actually performs the quantification (if using the *Run MassChroQ*) or the way *i2MassChroQ* writes the `masschroqml` file to be fed to the *MassChroQ* program.

This tab configures the way either *MassChroQ* actually performs the quantification or *i2MassChroQ* writes the `masschroqml` file that *MassChroQ* will be fed with to perform the task.

FIGURE 7.9: THE *MASSCHROQ* INTERFACE WINDOW (MASSCHROQ)

- *Edit MassChroQ execution*: activate the check button to use the directory icon to locate the *MassChroQ* program on disk. The full path to the program will be printed in the line edit widget next to the icon.

- *Run MassChroQ through HTCondor*: activate the check button to set the memory requirements for HT-Condor.

- *MassChroQ parameters*: these settings govern the actual *MassChroQ* quantification process:

  - *Number of CPUs*: set the number of central processing units that *MassChroQ* is allowed to use (these are actually called "threads").

  - *Temporary directory*: use the directory icon to select a specific temporary directory where *Mass-ChroQ* will write processing-related data. By default the directory is `/tmp/`. The temporary files are eliminated when no more used.

  - *Use the temporary directory to store detected peaks*: if checked, the detected peaks might be stored in files in the temporary directory described above. This can be construed as a swap area where to store peaks data if the available memory is insufficient.

Configuration of MassChroQ   i2MassChroQ 1.0.4

7.1.2.5 SAVING THE FILE AND OPTIONALLY RUNNING *MassChroQ*

Once all the configuration has been done, the user can either only save the `masschroqml` file by clicking on the *Save File* button or immediately start *MassChroQ* by clicking on the *Run MassChroQ* button.

> **NOTE**
>
> Even if the user decides to go down the direct *Run MassChroQ* route, the program will ask to save the `masschroqml` file. This is because that file is read by *MassChroQ* when *i2MassChroQ* internally calls it to run the quantification process.

The `masschroqml` file describes the proteins and peptides that were retained during the protein identification results analysis session. The contents of the file are shown in FIGURE 7.10, "CONTENTS OF THE MASSCHROQML FILE".

```
<rawdata>
  <data_file id="msruna1" format="mzml" path="1_A01_urnb-1.mzML" type="centroid"/>
  <data_file id="msruna2" format="mzml" path="1_A05_urnb-2.mzML" type="centroid"/>
  <data_file id="msruna3" format="mzml" path="1_A09_urnb-3.mzML" type="centroid"/>
  <data_file id="msruna4" format="mzml" path="1_B01_urnb-4.mzML" type="centroid"/>
  <data_file id="msruna5" format="mzml" path="1_B05_urnb-5.mzML" type="centroid"/>
</rawdata>

<groups>
  <group id="All_samples" data_ids="msruna1 msruna2 msruna3 msruna4 msruna5"/>
</groups>

<protein_list>
  <protein id="a1.a1.a1" desc="P04711 Phosphoenolpyruvate carboxylase"/>
  <protein id="b88.a1.a1" desc="NP_001152746 ascorbate peroxidase"/>
  <protein id="b88.a2.a1" desc="NP_001150192 APx1 - Cytosolic Ascorbate Peroxidase"/>
  <protein id="c388.a1.a1" desc="B6SIF9 Histone H2B seq=translation"/>

</protein_list>

<peptide_list>
  <peptide id="pepa1a1" mods="[C 43 H 75 O 14 N 12 S 0] " prot_ids="a1.a2.a1" mh="983.5520215842" seq="ADVLHLSTK">
    <observed_in data="msruna3" scan="5978" z="2"/>
  </peptide>
  <peptide id="pepa1a2" mods="[C 69 H 97 O 18 N 16 S 1] " prot_ids="a1.a3.a1 a1.a2.a1" mh="1469.6881979650" seq="AGMSYFHETIWK">
    <observed_in data="msruna5" scan="10834" z="3"/>
    <observed_in data="msruna4" scan="10880" z="3"/>
  </peptide>
  <peptide id="pepa1a3" mods="[C 55 H 98 O 18 N 15 S 0] " prot_ids="a1.a1.a1" mh="1256.7208778183" seq="AIADGSLLDLLR">
    <observed_in data="msruna1" scan="15046" z="2"/>
    <observed_in data="msruna4" scan="15130" z="2"/>
    <observed_in data="msruna2" scan="15300" z="3"/>
  </peptide>
  <peptide id="pepa1a4" mods="[C 74 H 121 O 26 N 18 S 1] " prot_ids="a1.a2.a1" mh="1709.8414637049" seq="ALLDEMAVVATEEYR">
    <observed_in data="msruna4" scan="14704" z="2"/>
    <observed_in data="msruna3" scan="14782" z="2"/>
    <observed_in data="msruna4" scan="14667" z="3"/>
    <observed_in data="msruna3" scan="14685" z="3"/>
    <observed_in data="msruna2" scan="14726" z="3"/>
  </peptide>
  <peptide id="pepa1a5" mods="[C 73 H 105 O 17 N 18 S 0] " prot_ids="a1.a1.a1" mh="1505.7899604389" seq="AIPWIFSWTQTR">
    <observed_in data="msruna1" scan="15553" z="2"/>
    <observed_in data="msruna2" scan="15687" z="2"/>
    observed_in data="msruna3" scan="15715" z="3"/
  </peptide>
</peptide_list>

<alignments>
    <alignment_methods>
        <alignment_method id="my_ms2">
            <ms2>
                <!--write_time_values_output_dir="directory" to write retention time corrections-->
                <ms2_tendency_halfwindow>10</ms2_tendency_halfwindow>
                <ms2_smoothing_halfwindow>15</ms2_smoothing_halfwindow>
                <ms1_smoothing_halfwindow>0</ms1_smoothing_halfwindow>
            </ms2>
        </alignment_method>
    </alignment_methods>
    <align group_id="All_samples" method_id="my_ms2" reference_data_id="msruna1"/>
</alignments>
```

The `masschroqml` file contains all the required data and configuration bits to perform the XIC extractions for all the peptidic precursor ions that allowed identifying proteins. This file is read by the *MassChroQ* program. (In this screen dump, the file contents were obviously redacted for brevity.)

FIGURE 7.10: **CONTENTS OF THE MASSCHROQML FILE**

7.2 INTERFACE TO THE *MSstats* STATISTICS MODULE

The statistical analysis of the quantified peptide data by *MassChroQ* is assigned to the *MSstats* software authored by M. Choi and colleagues (2014) *MSstats: An R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments* in *Bioinformatics*. As a prerequisite, peptide quantification must thus have been performed by *MassChroQ* as described in earlier sections.

This section describes in a step-by-step fashion the interface that *i2MassChroQ* provides to the *MSstats* software. To start the process, select the the *MSstats* menu item of the main *File* menu, as shown in FIGURE 7.11, "MSSTATS MENU ITEM IN THE MAIN I2MASSCHROQ MENU".
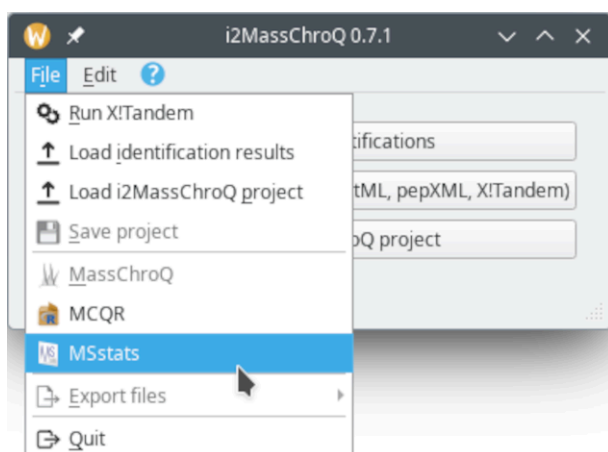
Menu that loads the interface to the *MSstats* statistics software that processes the *MassChroQ*-quantified peptide data to provide protein quantifications.

FIGURE 7.11: *MSSTATS* MENU ITEM IN THE MAIN *I2MASSCHROQ* MENU

When the *MSstats* interface is loaded it looks like that shown in FIGURE 7.12, "MAIN MSSTATS INTERFACE WINDOW". The actions to be carried over are shown on the left hand side region of the window in the form of a series of actions that are materialized by gradient-filled buttons. Each one of these actions are described in the following sections.
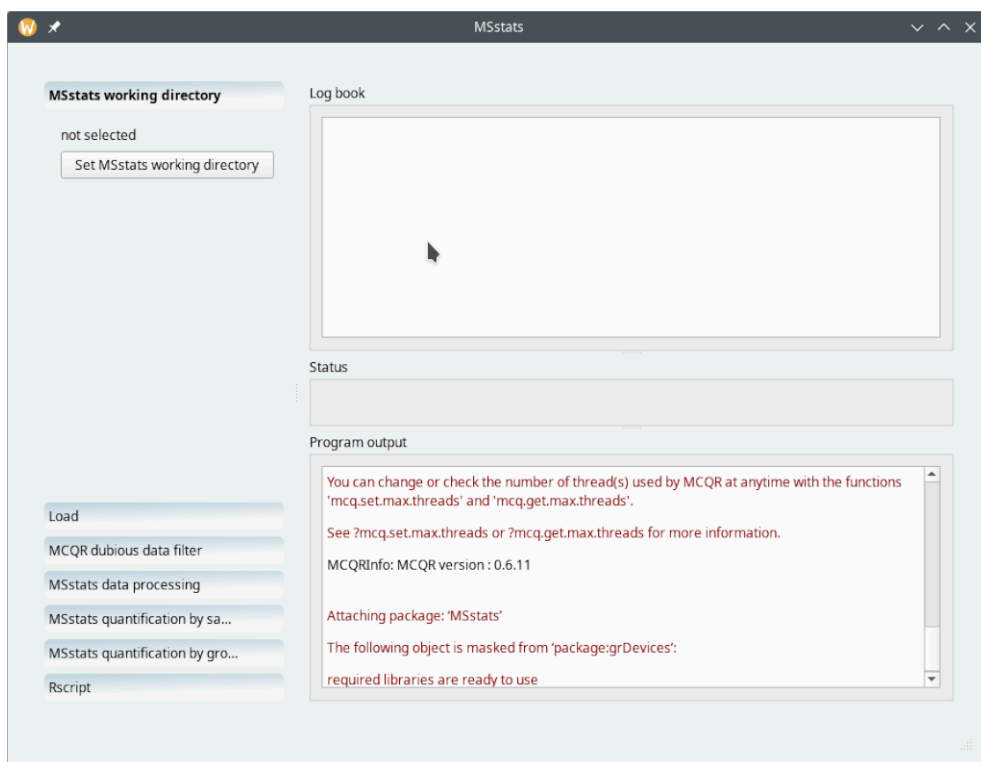
The main *MSstats* interface window has two main regions: the left hand side part of the window contains all the workflow steps that are to be carried out from the topmost item to the bottommost item; the right hand side part of the window contains three elements: the *Log book* view at the top, the *Program output* at the bottom, and the central *Status* widget.

FIGURE 7.12: MAIN *MSSTATS* INTERFACE WINDOW

7.2.1 SETTING THE TEMPORARY *MSstats* WORKING DIRECTORY

The first choice to be made by the user is to define where the *MSstats* package will write the data it needs to fulfill its statistical analysis tasks (FIGURE 7.13, "SETTING THE MSSTATS WORKING DIRECTORY"). The directory needs to be created if it does not exist already.
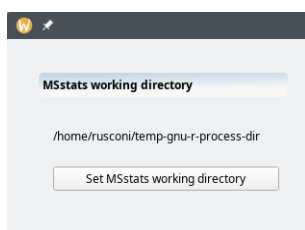
The directory is used by *MSstats* to store all the files and directories it creates during its fulfilling of its tasks. That directory can be located anywhere on the disk and needs to be created if it does not exist already.

FIGURE 7.13: SETTING THE *MSSTATS* WORKING DIRECTORY

7.2.2 LOADING THE PEPTIDE QUANTIFICATION DATA FILE BY *MassChroQ*

The *MassChroQ*-based peptide quantification process must have been performed already and must have produced an XML file with the `masschroqml` extension. That file can be loaded by clicking the *Select masschroqML file* button. Upon completing the data file loading process, the right hand side pane of the windows shows a summary of the data just loaded in the *Log book* widget (FIGURE 7.14, "LOADING MASSCHROQ-GENERATED DATA FILE").
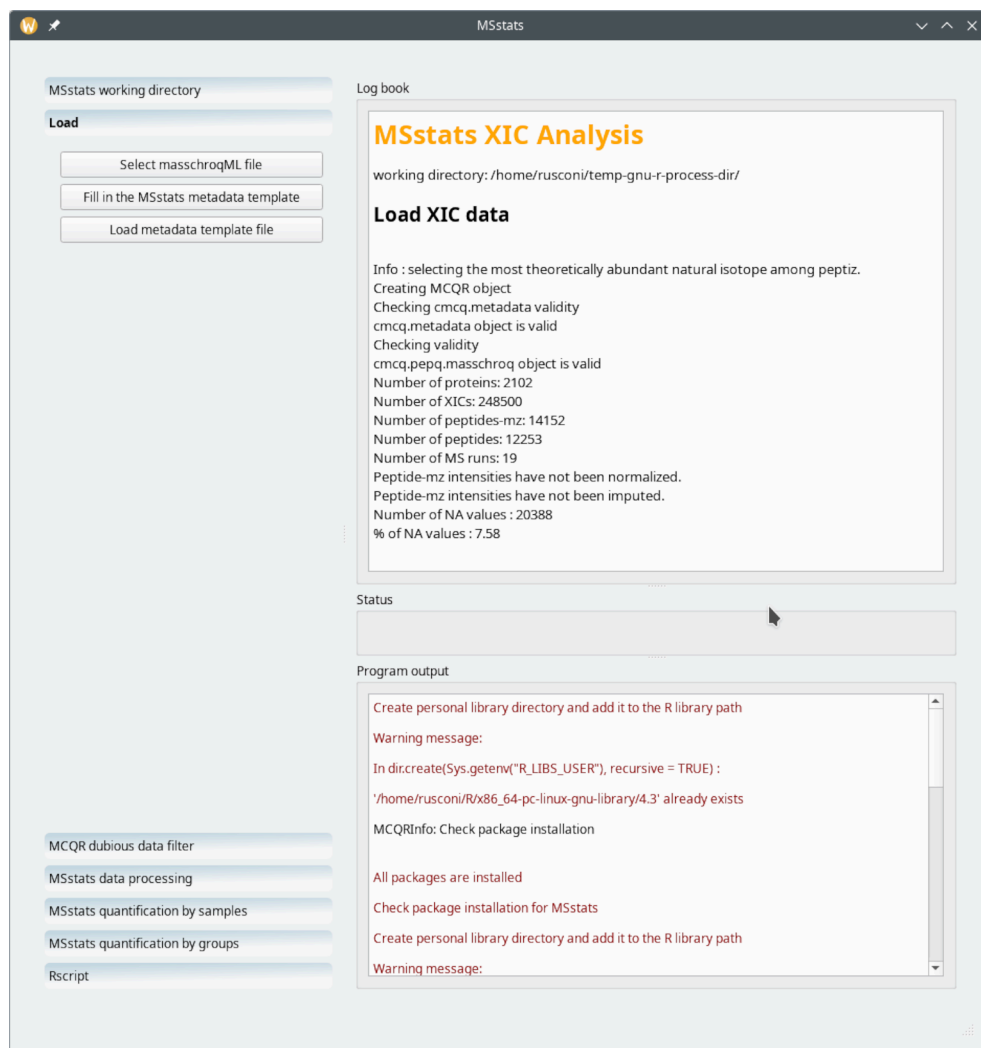
The peptide quantification process by *MassChroQ* produces a file that the user must load by clicking onto the *Select masschroqML file* button. As shown in the *Log book* pane on the right hand side of the window, the data have been loaded and a summary is provided.

FIGURE 7.14: LOADING *MASSCHROQ*-GENERATED DATA FILE

Once the data have been loaded, *i2MassChroQ* crafts a brand new spreadsheet data set in memory that needs to be stored on disk. For this, the user clicks the *Fill in the MSstats metadata template* button, which will permit saving the file to disk (with the OpenDocument format, `ods` extension, typically in the working directory created

earlier). *i2MassChroQ* will try to automatically open that file right after having written it to disk. If the file cannot be opened, then user needs to open it manually and start filling in the metadata for *MSstats* to performs its work as intended. A typical unmodified metadata template file looks like shown in Figure 7.15, "Metadata template file for use by MSstats" where only the MS run file names are listed.

A typical *MSstats* metadata template file as created by *i2MassChroQ*. That file needs to be modified according the user requirements in terms of grouping the samples according to the experiment plan. When created, the file only contains the list of MS runs from which peptide quantifications were performed.

**FIGURE 7.15: METADATA TEMPLATE FILE FOR USE BY *MSSTATS***

Once modified by the user to annotate the MS run file names and to group the samples according to the experiment plant, the file looks like shown in Figure 7.16, "Annotated metadata for use by MSstats".

	A	B	C	D
1	msrun	msrunfile	Condition	BioReplicate
2	msruna1	20120906_balliau_extract_1_A01_urnb-1.mzXML	urnb	1
3	msruna2	20120906_balliau_extract_1_A02_urzb-1.mzXML	urzb	2
4	msruna3	20120906_balliau_extract_1_A04_teal-1.mzXML	teal	3
5	msruna4	20120906_balliau_extract_1_A05_urnb-2.mzXML	urnb	4
6	msruna5	20120906_balliau_extract_1_A06_urzb-2.mzXML	urzb	5
7	msruna6	20120906_balliau_extract_1_A07_tca1-2.mzXML	tca	6
8	msruna7	20120906_balliau_extract_1_A08_teal-2.mzXML	teal	7
9	msruna8	20120906_balliau_extract_1_A09_urnb-3.mzXML	urnb	8
10	msruna9	20120906_balliau_extract_1_A10_urzb-3.mzXML	urzb	9
11	msrunb10	20120906_balliau_extract_1_A11_tca1-3.mzXML	tca	10
12	msrunb11	20120906_balliau_extract_1_A12_teal-3.mzXML	teal	11
13	msrunb12	20120906_balliau_extract_1_B01_urnb-4.mzXML	urnb	12
14	msrunb13	20120906_balliau_extract_1_B02_urzb-4.mzXML	urzb	13
15	msrunb14	20120906_balliau_extract_1_B03_tca1-4.mzXML	tca	14
16	msrunb15	20120906_balliau_extract_1_B04_teal-4.mzXML	teal	15
17	msrunb16	20120906_balliau_extract_1_B05_urnb-5.mzXML	urnb	16
18	msrunb17	20120906_balliau_extract_1_B06_urzb-5.mzXML	urzb	17
19	msrunb18	20120906_balliau_extract_1_B07_tca1-5.mzXML	tca	18
20	msrunb19	20120906_balliau_extract_1_B08_teal-5.mzXML	teal	19

Once the template metadata have been completed to inform *MSstats* about the sample grouping and the analysis logic, it might look like shown in this figure.

FIGURE 7.16: ANNOTATED METADATA FOR USE BY *MSSTATS*

The template file, as modified by the user should next be loaded by clicking on the *Load metadata template file* button. The *Log book* widget now shows informational data like shown in FIGURE 7.17, "PRELIMINARY PROCESSING PERFORMED BY MCQR UPON LOADING OF THE METADATA TEMPLATE FILE". The output data are comprehensive and illustrated with graphs like the one described below.
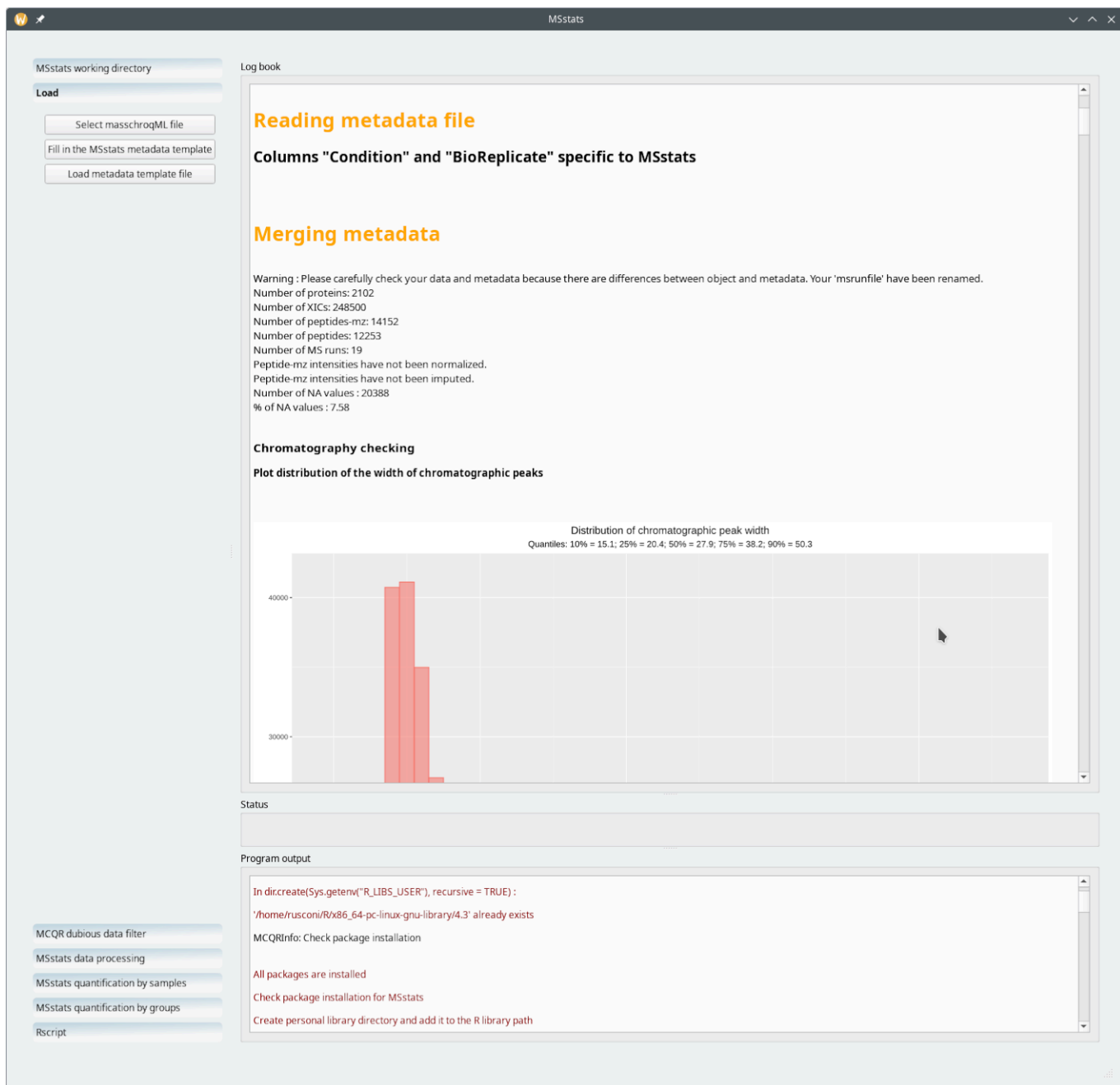
The template metadate file loading step triggers preliminary computing tasks by MCQR (a *GNU R* script developed in our facility) and the output is provided in the *Log book* widget.

FIGURE 7.17: PRELIMINARY PROCESSING PERFORMED BY MCQR UPON LOADING OF THE METADATA TEMPLATE FILE

One particular informational bit that is of use in a later step of the processing workflow is the distribution of the chromatographic peak width over all the samples and over the whole retention time (Figure 7.18, "Chromatography data checks: the distribution of the chromatography widths"). Equally useful is the distribution of retention time variability for all the (m/z,z) pairs that were extracted from the whole set of MS run acquisitions Figure 7.19, "Chromatography data checks: the distribution of retention time variations".
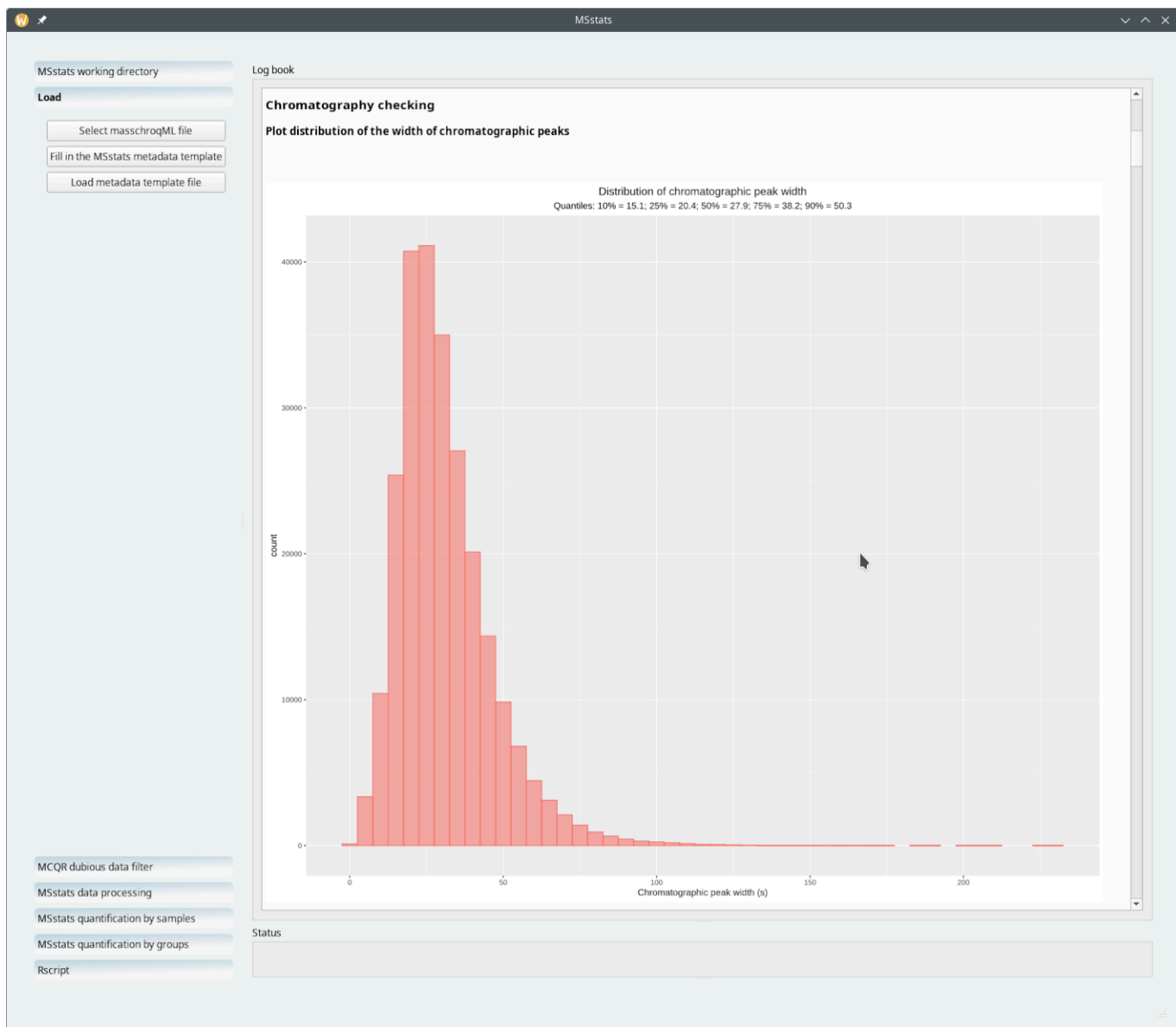
MSstats prints out data used by *i2MassChroQ* to plot a number of graphics like this histogram showing the distribution of the peak widths (in seconds). One can assume that a given ion might be reasonably contained in a retention time range (0—150) seconds.

**FIGURE 7.18: CHROMATOGRAPHY DATA CHECKS: THE DISTRIBUTION OF THE CHROMATOGRAPHY WIDTHS**
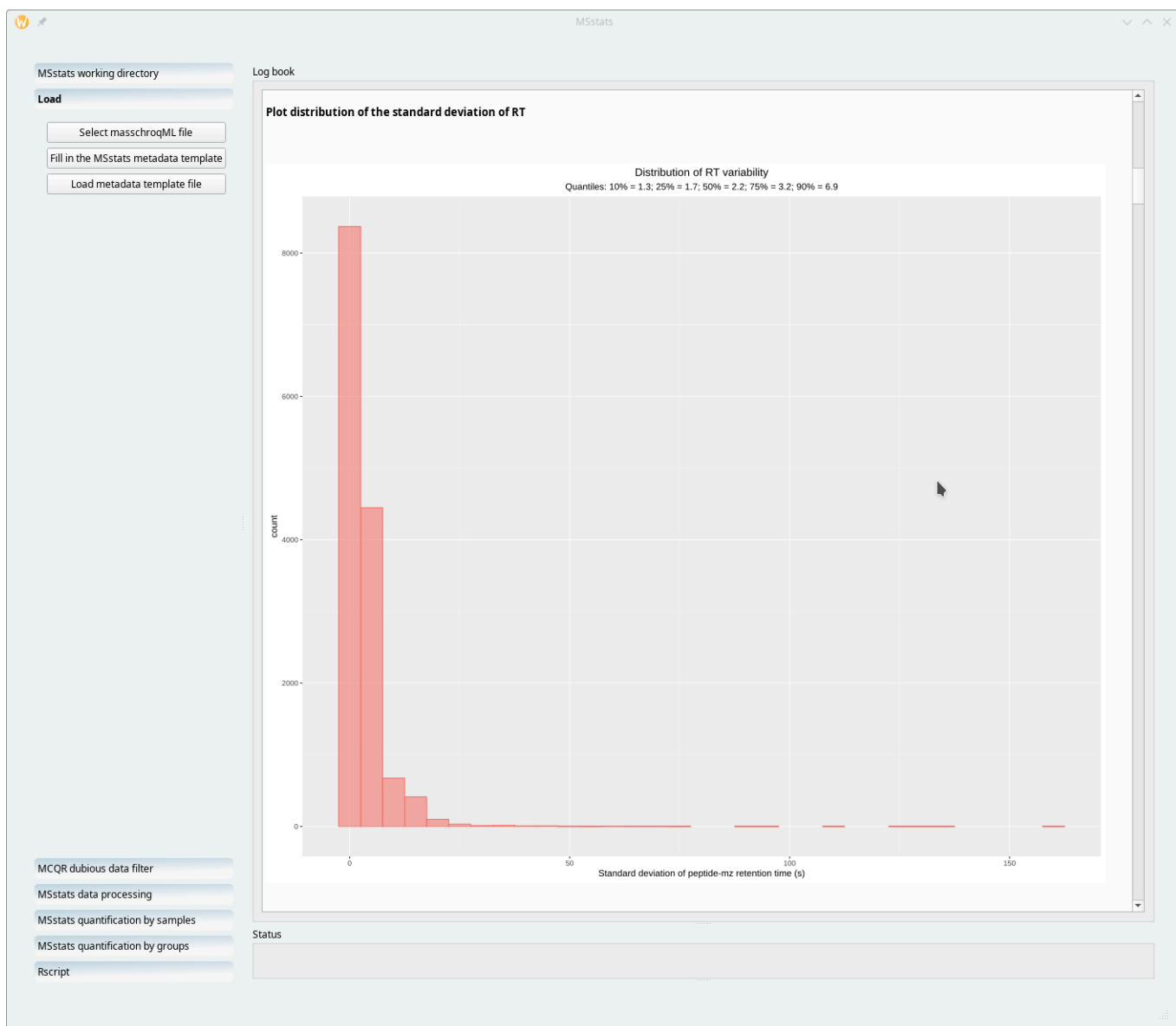
The histogram plot here shows the variability of the retention time values for all the (m/z,z) pairs extracted from the whole set of MS run acquisitions. One might consider that the maximum standard peak width variation acceptable is 30.

FIGURE 7.19: **CHROMATOGRAPHY DATA CHECKS: THE DISTRIBUTION OF RETENTION TIME VARIATIONS**

The two information bits described in the two figures above are of use in the next step of the processing workflow, as described in the next section.

7.2.3 FILTERING DUBIOUS DATA BY RUNNING MCQR

This step is optional. It is performed by MCQR. The idea is that the user should be able to scrap some dubious data from the data set if these data are outside of "reasonable ranges". For example, one should be able to filter out (m/z,z) pairs if they match retention times of too large a range (that is, for example, an ion being detected in the MS run acquisition over too long a retention time range, which is suspect).
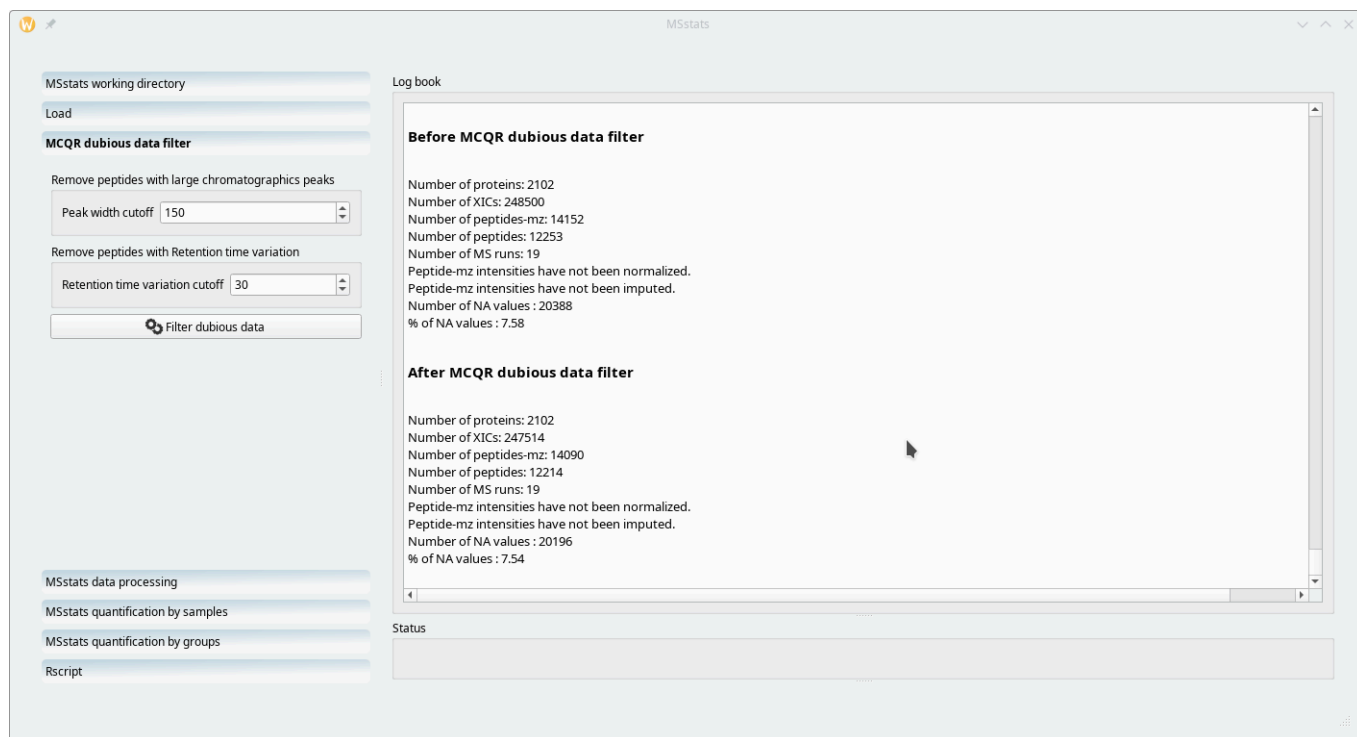
Dubious data might be filtered out on the basis of the two criteria shown. The numerical values set in this example are based on the output of *MSstats* as shown in the previous histograms.

FIGURE 7.20: FILTERING DUBIOUS DATA USING MCQR

The dubious data filtering is performed on the basis of two criteria: the retention time peak width (*Peak width cutoff*) and the retention time variability (*Retention time variation cutoff*). *i2MassChroQ* documents the filtering step in the *Log book* widget as shown in FIGURE 7.20, "FILTERING DUBIOUS DATA USING MCQR".

As visible in FIGURE 7.20, "FILTERING DUBIOUS DATA USING MCQR", in the ouput printed in the *Log book* widget, the data set is pretty good, since applying the filters did only remove 39 peptides over more than twelve thousands and not a single protein.

7.2.4 RUNNING *MSstats* ON THE CONFIGURED DATA SET

The next step in the wokflow is to actually run *MSstats*. However, one last bit of configuration is required: the user is requested to select the log transformation (either log2 or log10) because that is a prerequisite for *MSstats* to run. Once that configuration bit has been set, the user might click on the *Run MSstats* button.
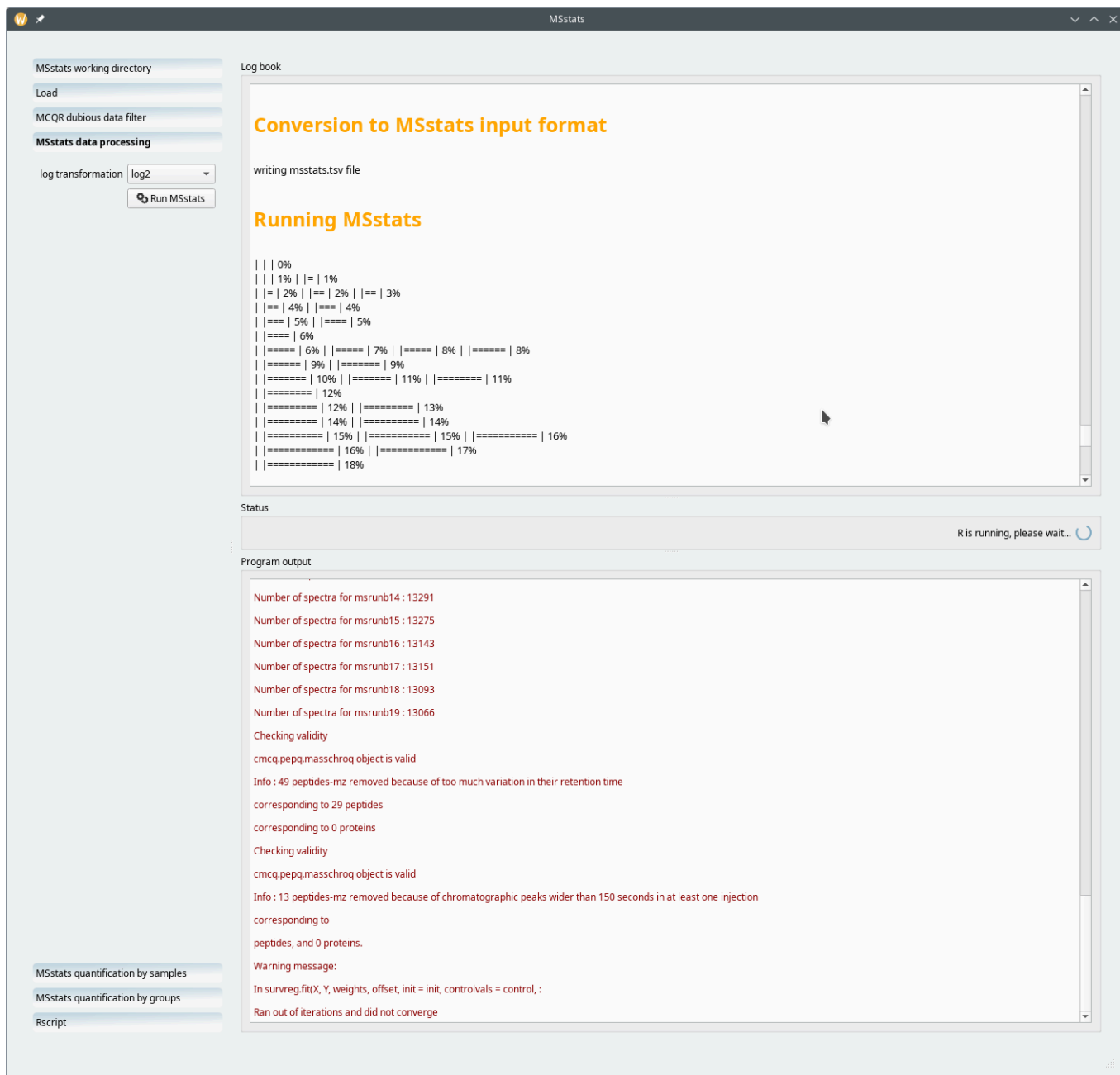
After having chosen the log tranformation (log2 or log10) that is *required* by *MSstats*, the user clicks on the *Run MSstats* button. The output shows the advancement of the computations.

FIGURE 7.21: RUNNING *MSSTATS* ON THE CONFIGURED DATA SET

At the end of the computation, as shown in FIGURE 7.21, "RUNNING MSSTATS ON THE CONFIGURED DATA SET", the starts the next workflow step.

7.2.5 Running the *MSstats* quantification by samples

One of the manners in which the *MSstats*-based quantification process can be run is the "quantification by samples" mode. This is triggered by clicking on the *MSstats quantification by samples* workflow item, as shown in Figure 7.22, "MSstats quantification by samples mode".
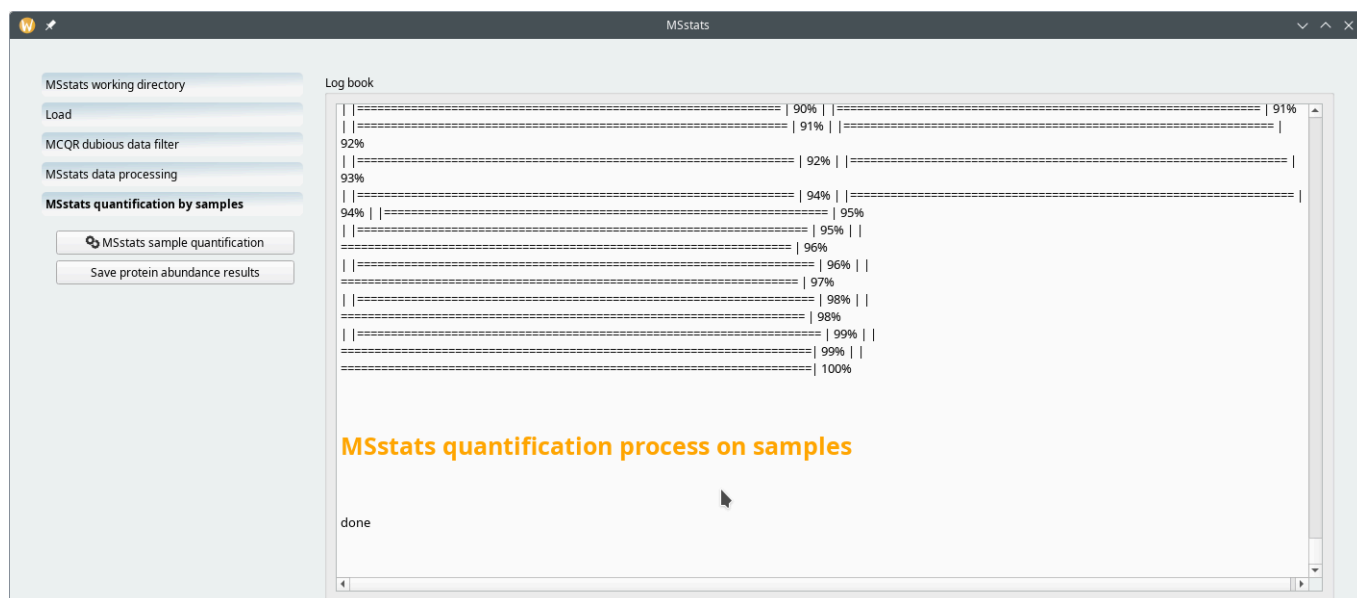
In the quantification by samples mode, the samples are taken as individual samples depending on their *BioReplicate* number in the metadata template file. See text for details.

FIGURE 7.22: *MSSTATS* QUANTIFICATION BY SAMPLES MODE

The quantification process depicted in Figure 7.22, "MSstats quantification by samples mode" is very quick. The "processing by samples" mode will quantify protein on the basis of the "*BioReplicate*" variable in the metadata template file (Figure 7.16, "Annotated metadata for use by MSstats"). Because, in the example, each MS run acquisition (that is, each row of the spreadsheet page) is marked as a different *BioReplicate*, the quantification by samples mode will quantify proteins found in each individual sample separately.

The user will want to save the protein quantification results by saving them to a spreadsheet file. This step is achieved by clicking the *Save protein abundance results* button. The saved spreadsheet file is shown in Figure 7.23, "Spreadsheet view of the quantification by samples results".

In the metadata template file, each MS run acquisition was listed as a different *BioReplicate* identity. This means that proteins were quantified independently in each MS run. The spreadsheet view in this figure shows quantification data for each protein (each row) found in each each sample (the columns).

FIGURE 7.23: SPREADSHEET VIEW OF THE QUANTIFICATION BY SAMPLES RESULTS

7.2.6   RUNNING THE *MSstats* QUANTIFICATION BY GROUPS

The other manner in which the *MSstats*-based quantification process can be run is the "quantification by groups" mode. This is triggered by clicking on the *MSstats quantification by groups* workflow item, as shown in FIGURE 7.24, "MSSTATS QUANTIFICATION BY GROUPS MODE".
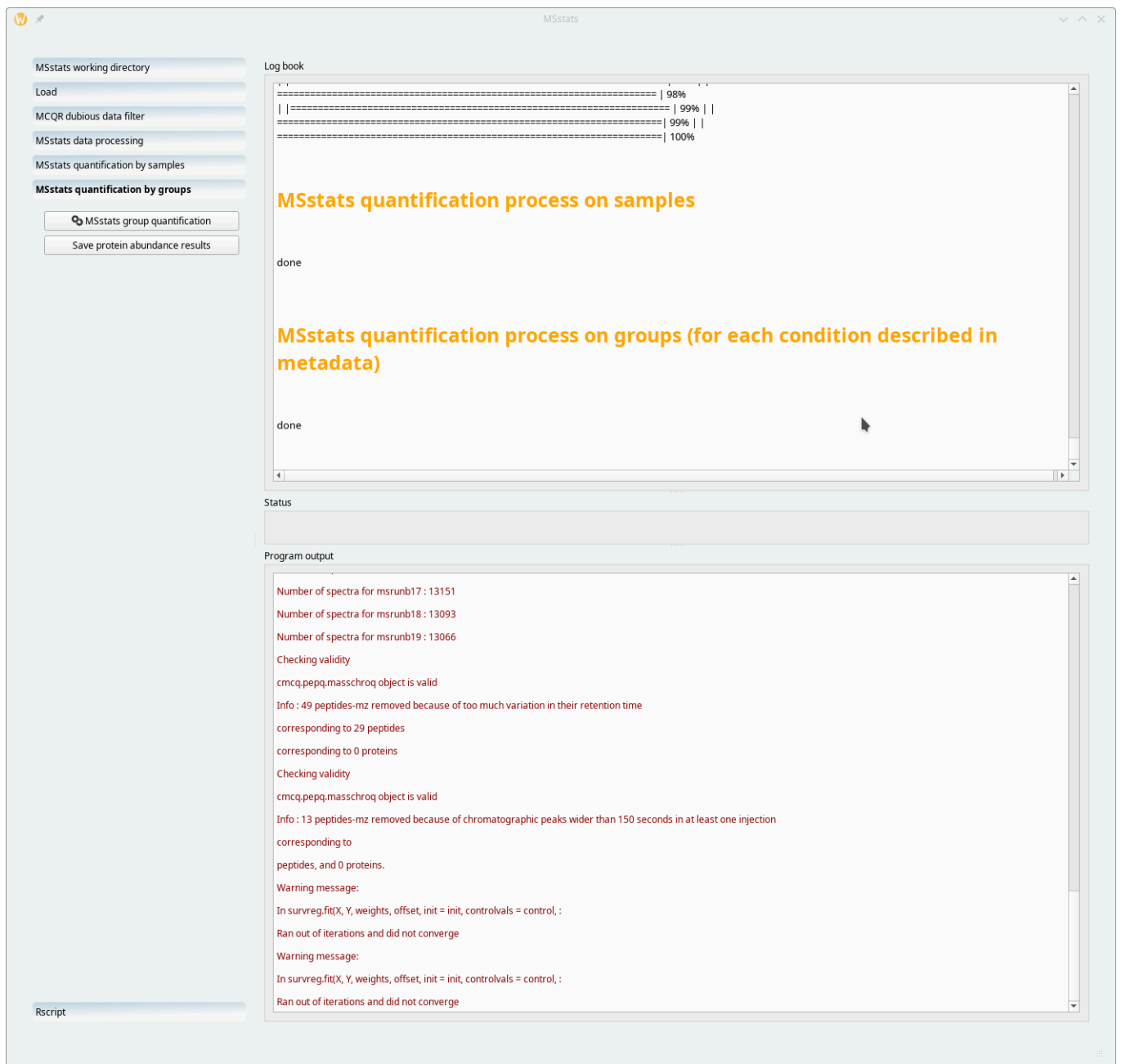
In the quantification by groups mode, the samples are first grouped into groups defined in the metadata template file. In that file, the column that specifies the required grouping has the *Condition* header. See text for details.

FIGURE 7.24: *MSSTATS* QUANTIFICATION BY GROUPS MODE

The quantification process depicted in Figure 7.24, "MSstats quantification by groups mode" is very quick. The "processing by groups" mode will quantify proteins on the basis of the "*Condition*" variable in the metadata template file (Figure 7.16, "Annotated metadata for use by MSstats"). Because, in the example, there are four different *Condition* values, there will be four groups of proteins (different protein solubilization methods).

The user will want to save the protein quantification results by saving them to a spreadsheet file. This step is achieved by clicking the *Save protein abundance results* button. The saved spreadsheet file is shown in Figure 7.25, "Spreadsheet view of the quantification by groups results".

In the metadata template file, MS run acquisitions were grouped using the value of the *Condition* variable. The grouping of the MS runs involve thus four groups (different protein solubilization methods). This means that the proteins will be quantified in each group of MS run acquisitions.

FIGURE 7.25: **SPREADSHEET VIEW OF THE QUANTIFICATION BY GROUPS RESULTS**

7.2.7 Running the *MSstats GNU R and RMarkdown scripts*

The workflow as has developed since the beginning to this *MSstats* work session has been recorded both in the form of a pure *GNU R* script and as a RMarkdown script. By clicking onto the *Rscript* workflow item (Figure 7.26, "Load the GNU R and RMarkdown scripts"), the user is presented with two options: load the *GNU R* script and/or the RMarkdown script. The RMarkdown script might be run in the *RStudio* (*RStudio* has changed its name to become *Posit*) environment.
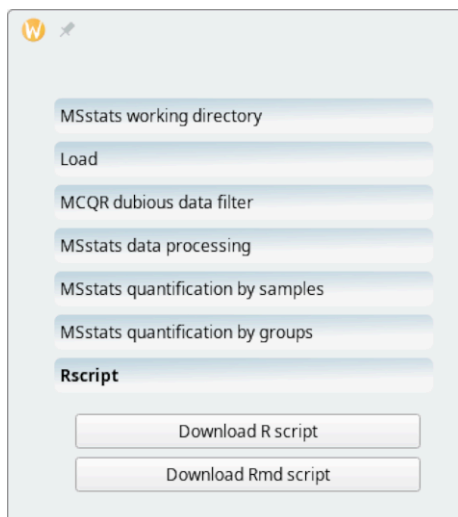
Use the button of interest to download the *GNU R* or the RMarkdown script corresponding to all the workflow steps that were run up to this one.

FIGURE 7.26: LOAD THE *GNU R* AND RMARKDOWN SCRIPTS

Upon saving the RMarkdown version of the script, if available on the system, *i2MassChroQ* will try to load it automatically in *RStudio*.

8 Specific procedures for the timsTOF line of instruments

This chapter describes the very few specific procedures to carry out when the proteomics data at hand are from the Bruker timsTOF line of instruments.

8.1 General considerations

The mass spectrometers from the timsTOF line of instruments by Bruker implement ion mobility mass spectrometry by trapping ions and subjecting them to a gas flow that moves them in the trap according to their collisional cross section. Interestingly, the outcome of that operation is the reverse of conventional drift tube-based ion mobility: large ions are released first and smaller ions are released last.

Apart from the observation above, the result is nonetheless that ions entering the instrument are separated according to a new dimension that is orthogonal to the other two retention time and m/z ratio dimensions: the ion mobility dimension. This new dimension inevitably introduces more complexity and greater volume to the mass data.

In a historic move, Bruker has decided to publish the technical details of their data format. During the acquisition, data are stored in two separate files located in their data directory that has the `.d` extension:

- `analysis.tdf`: this file is a *SQLite3* relational database that contains all the metadata about the acquisition. The generic *metadata* term defines data that describe the actual data. So this file contains data that explain how the data are organized in the actual data file below.

- `analysis.tdf_bin`: this file is a binary-format file that holds the data in the form of a succession of numbers packed according to a specific scheme that Bruker has decided to make public.

The *SQLite3* `analysis.tdf` file contains a set of tables. Most often, records in one table make reference (*relate*) to other record(s) in other table(s). This is why this database file is said to be a *relational database*. A view of the tables making that relational database file is shown in Figure 8.1, "View of the relational database file".
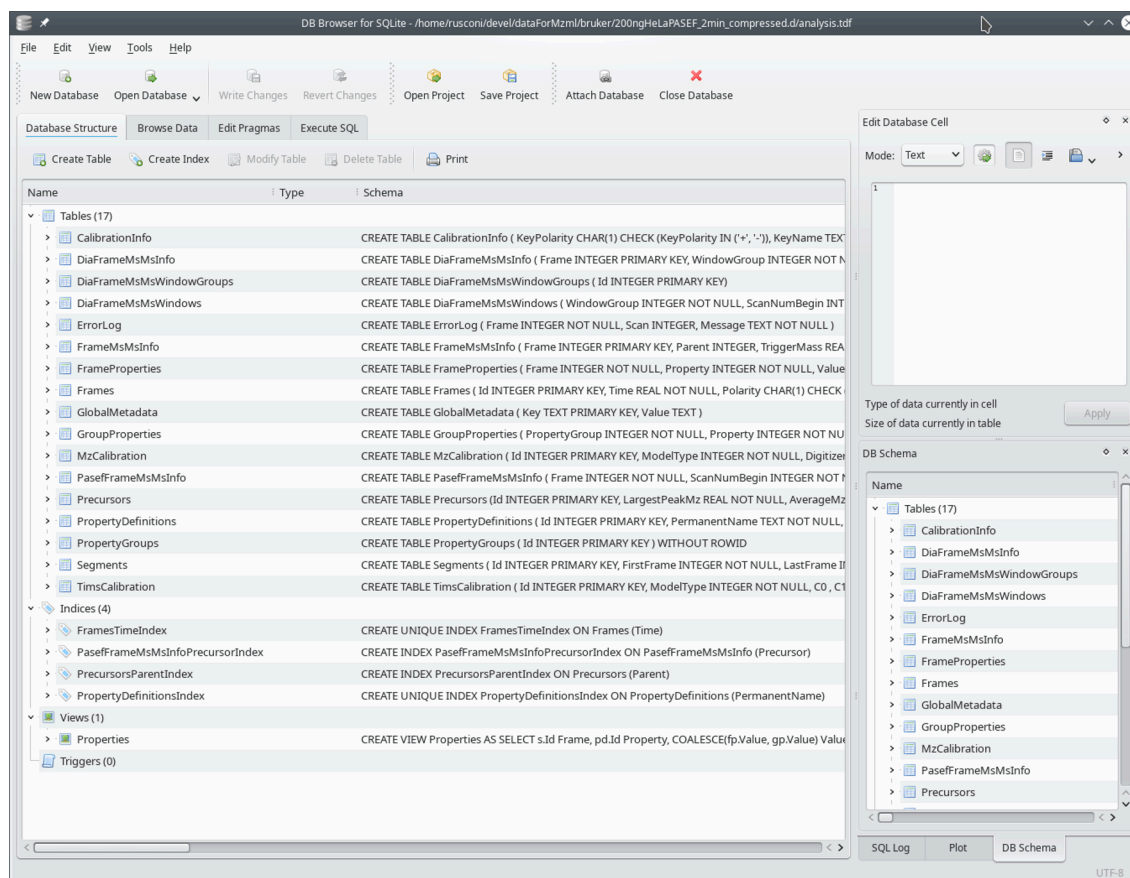
The mass spectrometers of the timsTOF line of instruments by Bruker produce mass data that are stored in two files. This figure shows the table structure of the relational database `analysis.tdf` file displayed in *Sqlite-Browser* (a Free Software application).

FIGURE 8.1: VIEW OF THE RELATIONAL DATABASE FILE

When dealing with proteomics projects that have their data originating in timsTOF instruments, some specific steps are to be taken so as to inform *i2MassChroQ* that specific handling is required. These will be reviewed below in the same succession as they need be implemented when running *i2MassChroQ*.

8.1.1   RUNNING *X!TANDEM* IDENTIFICATIONS WITH BRUKER TIMSTOF DATA

It is possible to load Bruker timsTOF data right in the *i2MassChroQ* program's graphical user interface, as shown in the window pictured in FIGURE 3.2, "X!TANDEM-BASED IDENTIFICATION CONFIGURATION". The very first specific step to take in this case is to select the data files by clicking the *Add Bruker timsTOF folders*.

The *Add Bruker timsTOF folders* button lets the user choose the Bruker data directory (`.d` extension) and then asks if the data to be loaded are in the TDF or MGF format (see FIGURE 8.2, "FILE FORMAT SELECTION DIALOG FOR BRUKER TIMSTOF DATA").

Indeed, the MGF file generated by the Bruker software is automatically installed in the `.d` extension data directory.
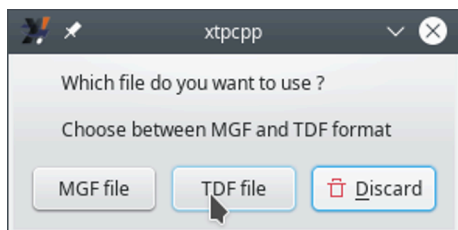
When handling Bruker timsTOF data, two file formats are available: MGF and TDF. See text for details.

FIGURE 8.2: FILE FORMAT SELECTION DIALOG FOR BRUKER TIMSTOF DATA

The *DataAnalysis* software from Bruker allows one to export proteomics MS/MS data into MGF format files (Mascot generic format). Their native data format, though, is the TDF format. It is important to keep in mind that the MGF format only stores MS/MS spectral data, no MS data. By using this format, *i2MassChroQ* and *MassChroQ* won't be able to access MS data, which are required in a number of situations, in particular when extracting ion currents for given m/z ratios, for example, or for area-under-the-curve quantitative proteomics. It is thus always recommended to use the native TDF format whenever available.

When loading Bruker timsTOF data right into *i2MassChroQ* as described above, the software performs some under-the-hood operations that the user might want to be aware of. The hidden operations are unveiled in the following sections, as they involve command line programs shipped along with *i2MassChroQ* that might be of interest to the user.

8.1.2 Converting Bruker timsTOF data to mzXML with mzxmlconverter

X!Tandem needs the mass spectrometric data that it uses for the database searches to be in the mzXML format. For this very reason, *i2MassChroQ* cannot work by harnessing the capabilities of *X!Tandem* starting from Bruker timsTOF data. These data need to be converted to an mzXML file before they can then be fed to *X!Tandem*.

In order to be able to store a mzXML file on disk, the user may convert timsTOF data files (TDF or MGF) to mzXML using the *mzxmlconverter* program that is shipped along with *i2MassChroQ*. This program is a command line program that takes a data file in input and that writes a mzXML file in output. The command line syntax is easy, as picture in Figure 8.3, "Converting mass data files to mzXML data files".

To obtain help about the program, run the following:

$ [1] **`<path_to>/mzxmlconverter --help`** RETURN

---

1 The prompt character might be *%* in some shells, like *zsh*.

```
% src/mzxmlconverter --help
Usage: src/mzxmlconverter [options]
X!TandemPipeline
mzXML converter

mzxmlconverter converts mass spectrometry data files from any format understood by ProteoWizard or from Bruker's timsTOF format to an
 mzXML format that is understood by X!Tandem.

Options:
  -h, --help              Displays help on commandline options.
  --help-all              Displays help including Qt specific options.
  -v, --version           Displays version information.
  -o, --output <output>   Write mzXML output file <output>.
  -i <input>              m/z data file <input>.
  -c, --cpus <cpus>       number of CPUs to use <cpus>.
  -d, --dalton <dalton>   timsTOF MS2 measurement precision in dalton (+-0.02 by
                          default).
```

Files of any format handled by *ProteoWizard* or files from the Bruker's timsTOF line of instruments can be converted to mzXML using *mzxmlconverter*. Conversion from timsTOF format to mzXML is performed entirely by our own software.

FIGURE 8.3: **CONVERTING MASS DATA FILES TO MZXML DATA FILES**

✋ Warning: The mzXML file format does not contain mobility data from the timsTOF data files

It is important to grasp that the mzXML file that is generated by *mzxmlconverter* does not contain all the mass data that are contained in the Bruker's timsTOF data files. When loading the produced mzXML format files in *i2MassChroQ*, the *X!Tandem* program will be able to perform peptide and protein identifications. Later, however, when the user will try to activate features in *i2MassChroQ* that require the original data, the software will not be able to provide the expected results, like XIC reports or ion mobility values, because it won't have access to the original data.

The *mzxmlconverter* program is practical because it allows storing the mzXML file on disk and loading it in *i2MassChroQ* for *X!Tandem* to consume it for the identifications. However, as stated in the warning above, that mzXML file has not a full copy of the data in the original mass spectrometry data file (be that a mzML, or MGF, or TDF file). *i2MassChroQ* has a solution for this problem: using an integrated workflow to convert the original data file to mzXML, make *X!Tandem* use it, write out the *X!Tandem* results file and finally rewrite that file into a new version by adding a connection between the *X!Tandem* run results and the original mass data file. In this way, when the user will activate features in *i2MassChroQ* that need accessing the original mass data file, the expected results will be effectively displayed. This process is described in detail in the next section.

8.1.3 Data conversion process with Bruker timsTOF TDF data and tandemwrapper

In order to seamlessly use Bruker timsTOF data in the context of performing *X!Tandem* identifications and later *MassChroQ*-based quantifications, the *tandemwrapper* program is made available to users who want to perform database searches using the command line interface. The *tandemwrapper* program performs an under-the-hood

file format conversion as described in the previous section before automatically feeding the generated mzXML file to *X!Tandem*. After *X!Tandem* has produced its results file, that file is rewritten by *tandemwrapper* in such a manner that the connection with the original mass spectrometry data files is reinstated for further use in the *i2MassChroQ* graphical interface.

There is, however, a way to convert Bruker timsTOF data files to mzXML using a standalone program, called *tandemwrapper*, that is shipped along with *i2MassChroQ*. The *tandemwrapper* program is a command line program that takes as input a XML configuration file. The XML file is most similar to the configuration file that *X!Tandem* uses.

To obtain help about the program, run the following:

$ **`<path_to>/tandemwrapper --help`** `RETURN`

A typical *tandemwrapper* input configuration file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<bioml label="example-tandemwrapper-mass-data-file.mzxml">
  <note type="heading">Paths</note>
  <note type="input" label="list path, default parameters">/full_path_to/xtandem-
presets-file.xml</note>
  <note type="input" label="list path, taxonomy information">/full_path_to/
database.xml</note>
  <note type="input" label="spectrum, path">full_path_to/mass-data-file.mzxml</note>
  <note type="heading">Protein general</note>
  <note type="input" label="protein, taxon">usedefined</note>
  <note type="heading">Output</note>
  <note type="input" label="output, path">/full_path_to/tandemwrapper-output.xml</note>
</bioml>
```

The XML configuration file that is provided to *tandemwrapper* on the command line is a replicate of the file that *X!Tandem* itself expects. That file is shown in FIGURE 8.4, "CONFIGURATION FILE FOR TANDEMWRAPPER".

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <bioml label="example-tandemwrapper-mass-data-file.mzxml">
3      <note type="heading">Paths</note>
4      <note type="input" label="list path, default parameters">/full_path_to/xtandem-presets-file.xml</note>
5      <note type="input" label="list path, taxonomy information">/full_path_to/database.xml</note>
6      <note type="input" label="spectrum, path">full_path_to/mass-data-file.mzxml</note>
7      <note type="heading">Protein general</note>
8      <note type="input" label="protein, taxon">usedefined</note>
9      <note type="heading">Output</note>
10     <note type="input" label="output, path">/full_path_to/tandemwrapper-output.xml</note>
11 </bioml>
```

The *tandemwrapper* program takes as input a configuration file that is most similar to the configuration file that is fed to *X!Tandem*.

FIGURE 8.4: **CONFIGURATION FILE FOR TANDEMWRAPPER**

The following elements need an explanation:

- *default parameters*: In the example, the `/full_path_to/xtandem-presets-file.xml` file is the *X!Tandem* presets file, already discussed in Section 3.3, "Setting the X!Tandem Run Presets".

- *taxonomy information*: In the example, the `/full_path_to/database.xml` file is the file that configures the location of the FASTA protein database files that are searched by *X!Tandem*. This file is described below.

- *path*: In the example, the `/full_path_to/mass-data-file.mzxml` file is the mass spectrometry data file in the mzXML format.

> **⊙ TIP**
>
> The mass spectrometry data file might be of any format that can be handled by ProteoWizard (open data formats only, particularly mzML) and also the Bruker's timsTOF TDF format that is handled by our own code.

- *output, path*: In the example, the `/full_path_to/tandemwrapper-output.xml` file is the file in which the *X!Tandem* configuration file is written for immediate use by *X!Tandem*.

The configuration file that indicates where the FASTA protein database files are located, that is referenced in the *tandemwrapper* input configuration file is shown in Figure 8.5, "Configuration file pointing at the FASTA protein databases".

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <bioml label="x! taxon-to-file matching list">
3   <taxon label="usedefined">
4     <file format="peptide" URL="/full_path_to/uniprot-coli_20211011.fasta"/>
5     <file format="peptide" URL="/full_path_to/uniprot-yeast_20211011.fasta"/>
6     <file format="peptide" URL="/full_path_to/uniprot-human-20211006.fasta"/>
7   </taxon>
8 </bioml>
9 
```

This file tells *tandemwrapper* the location of the FASTA protein databases required when *X!Tandem* will actually perform the searches.

FIGURE 8.5: CONFIGURATION FILE POINTING AT THE FASTA PROTEIN DATABASES

The *tandwrapper* program performs the following tasks in sequence:

- Convert the input mass data file to the mzXML file format that *X!Tandem* needs to perform the database searches. This step is only performed if the original mass spectrometry data file has not the mzXML format. *X!Tandem* produces an identification results file in an XML format;

- The mzXML format that is consumed by *X!Tandem* is a pretty simple format that was not designed to store a large variety of data/metadata, like ion mobility data, for example. For this very reason, *tandemwrapper* reads the identification results file produced by *X!Tandem* (that file is also in a specific

XML format) and rewrites it to a new analogous file that has all the necessary connections to the original mass data file. In this way, when the new version of the *X!Tandem* identification results file is loaded in *i2MassChroQ* all the original mass data can be accessed to provide the user with all the data, like XIC chromatograms, ion mobility data, for example. To load the *X!Tandem* identification results, see Section 3.4, "Loading the Protein Identification Results".

A GNU General Public License version 3

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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 HTTPS://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 HTTPS://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 HTTPS://WWW.GNU.ORG/LICENSES/WHY-NOT-LGPL.HTML ↗ .

Colophon

About the authors. Filippo Rusconi is a senior research scientist at the French national research council (*Centre national de la Recherche scientifique*, CNRS). Filippo has a background in biochemistry and organic chemistry and was trained during his Ph.D. as a bioanalytical chemist. He has extensive knowledge of analytical techniques involved in the study of biopolymers.

Filippo Rusconi is the author of a handbook about mass spectrometry for biochemists (French). The book was published by the French sci/tech publisher Lavoisier (https://www.lavoisier.fr) ↗ .

Colophon. The look of this book (PDF file) is the result of me having read many books from the O'Reilly publisher.

The frog on the book title page is a frog from Papua. This frog is able to hover when performing downwards leaps. This picture is courtesy http://www.papuaweb.org ↗

The typesetting of the book has been done on a Debian GNU/Linux computer using only Free Software. Use of the DocBook Authoring and Publishing Suite (DAPS (https://github.com/openSUSE/daps) ↗) from SUSE was key in the process.

The layout adopted for this book is an adaptation of the SUSE stylesheets. I would like to thank Frank Sundermeyer <fsundermeyer@opensuse.org> and Stefan Knorr <sknorr@suse.de> for being helpful with all my questions.

The main font used was EBGaramond (https://github.com/georgd/EB-Garamond) ↗ and the symbol/mathematical font was from the STIX project (https://www.stixfonts.org/) ↗ (font: STIX2Math).

The screen shots were taken with Spectacle, the screen capture program shipped along with my KDE (https://www.kde.org/) ↗ desktop environment and resampled using The GNU image manipulation program The Gimp (https://www.gimp.org/) ↗ . Illustrations were done in Inkscape (https://inkscape.org/) ↗ , a vectorial drawing software.