

I2MASSCHROQ USER MANUAL

FREE AND OPEN SOURCE PROTEIN IDENTIFICATION SOFTWARE

version: 1.2.0

Benoît Valot
Thomas Renne
Michel Zivy

Olivier Langella
Filippo Rusconi

CONTENTS

1	GENERALITIES	7
1.1	History of the project	7
1.2	What does <i>izMassChroQ</i> Stand for?	8
1.3	Transitioning from <i>X!TandemPipeline++</i> to <i>izMassChroQ</i>	8
1.4	General concepts and terminologies	8
1.5	Citing the <i>izMassChroQ</i> software.	10
1.6	Installation of the software	11
2	FUNDAMENTALS IN BOTTOM-UP PROTEOMICS	12
2.1	The Protein Biopolymer: Structure and Chemistry	12
2.2	General Overview of Bottom-up Proteomics	19
3	THE MAIN PROGRAM WINDOW	34
3.1	Starting a new <i>izMassChroQ</i> working session	34
3.2	Running <i>X!Tandem</i> identifications	35
3.3	Setting the <i>X!Tandem</i> Run Presets	36
3.4	Loading the Protein Identification Results	40
4	EXPLORING IDENTIFICATION DATA	47
4.1	The Protein List Window	47
4.2	The Peptide List Window	53
	BIBLIOGRAPHY	57

LIST OF FIGURES

FIGURE 2.1	PEPTIDIC BOND FORMATION BY CONDENSATION	13
FIGURE 2.2	END CAPPING CHEMISTRY OF THE PROTEIN POLYMER	14
FIGURE 2.3	PROTEIN CLEAVAGE BY WATER AND CYANOGEN BROMIDE	16
FIGURE 2.4	PROTEIN FRAGMENTATION PATTERNS MOST WIDELY ENCOUNTERED	18
FIGURE 2.5	THEORETICAL AND EXPERIMENTAL PARALLEL DATA-PRODUCING PROCESSES	20
FIGURE 2.6	THE STEPS LEADING TO A SCORED PEPTIDE <i>VS</i> MASS SPECTRUM MATCH (PSM)	25
FIGURE 2.7	COMPUTATION OF A PEPTIDIC EXPECTATION VALUE (E-VALUE)	29
FIGURE 2.8	PROTEIN INFERENCE: CONSTRUCTING A CONSOLIDATED PROTEIN IDENTIFICATIONS LIST	31
FIGURE 2.9	PHOSPHO-SITE INFERENCE: CONSTRUCTING A CONSOLIDATED PHOSPHO-SITE LIST	33
FIGURE 3.1	MAIN PROGRAM WINDOW	35
FIGURE 3.2	<i>X!TANDEM</i> -BASED IDENTIFICATION CONFIGURATION	35
FIGURE 3.3	<i>X!TANDEM</i> PRESETS CONFIGURATION WINDOW (<i>SPECTRUM</i> TAB)	37
FIGURE 3.4	<i>X!TANDEM</i> RUN FEEDBACK TO THE USER	39
FIGURE 3.5	<i>X!TANDEM</i> RUN FINISHED MESSAGE TO THE USER	39
FIGURE 3.6	CONFIGURATION OF THE LOADING OF THE IDENTIFICATION RESULTS	41
FIGURE 3.7	SELECTING A PARTICULAR IDENTIFICATION RESULTS FILE'S DATA SET	42
FIGURE 3.8	SELECTING A PARTICULAR IDENTIFICATION RESULTS FILE'S DATA SET	43
FIGURE 3.9	DISPLAYING THE MS IDENTIFICATIONS LIST (FIRST COLUMNS)	46
FIGURE 3.10	DISPLAYING THE MS IDENTIFICATIONS LIST (LAST COLUMNS)	46
FIGURE 4.1	THE PROTEIN LIST WINDOW	47
FIGURE 4.2	PROTEIN IDENTIFICATION FILTER PARAMETERS TAB OF THE MAIN WINDOW	49
FIGURE 4.3	FALSE DISCOVERY RATE (FDR) DATA AFTER A PROTEIN INFERENCE PROCESS IS RUN	50
FIGURE 4.4	MASS PRECISION QUALITY ASSESSMENT	51
FIGURE 4.5	PROTEIN DETAILS WINDOW	52
FIGURE 4.6	THE PEPTIDE LIST WINDOW (FIRST COLUMNS)	53
FIGURE 4.7	PEPTIDE LIST WINDOW (LAST COLUMNS)	53
FIGURE 4.8	COLUMNS THAT POPULATE THE PEPTIDE LIST TABLE VIEW	54
FIGURE 4.9	PEPTIDE DETAILS WINDOW	55

PREFACE

SOFTWARE FEATURE OFFERINGS AND INTENDED AUDIENCE

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

i2MassChroQ has the following features:

- Load mass spectrometry data files in the mzXML or mzML format, thanks to the excellent *libpwiz* library of ProteoWizard¹ 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.

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/](http://papps0.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/](http://proteowizard.sourceforge.net/)

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/](http://PAPPSO.INRAE.FR/EN/BIOINFO/XTANDEMPIPELINE/)². 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/XTANDEMPIPELINE/DOWNLOAD/](http://PAPPSO.INRAE.FR/EN/BIOINFO/XTANDEMPIPELINE/DOWNLOAD/)²) 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* philosophy, the reader is kindly urged to visit [HTTP://WWW.GNU.ORG/PHILOSOPHY](http://WWW.GNU.ORG/PHILOSOPHY)².

²[HTTP://WWW.DEBIAN.ORG/](http://WWW.DEBIAN.ORG/)

I 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 Langella, O., Valot, B., Balliau, T., Blein-Nicolas, M., Bonhomme, L., and Zivy, M. (2017). X!TandemPipeline: A Tool to Manage Sequence Redundancy for Protein Inference and Phosphosite Identification. *Journal of Proteome Research* 16, 494–503. doi: 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 stand-alone 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 *MCQR* project that was developed as a standalone project. *MCQR* 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 *MCQR* 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 identified peptides and inferred proteins. *i2MassChroQ* has area-under-the-curve quantitative proteomics capabilities 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 Chromatograms*” for the Quantification.

From the sequence above, the `&i2mcq;` name becomes self-explanatory!



TIP

It is however possible (and encouraged) to mentally read *i2MassChroQ* as “*I too Mass-ChroQ !*”

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 `PAPPS0/xtpcpp.conf` file. In order to preserve these configuration data after having transitioned from *X!TandemPipeline++* to *i2MassChroQ*, please, rename that configuration file to `PAPPS0/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 identifying 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 proteomics: “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 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 access 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 *MassChroQ* module to perform the bio-statistical analysis of the quantitative proteomics data obtained at the previous step.

1.5 CITING THE *i2MASSCHROQ* SOFTWARE.

Please cite the latest article :

Langella, O., Renne, T., Balliau, T., Davanture, M., Brehmer, S., Zivy, M., et al. (2024). Full Native timsTOF PASEF-Enabled Quantitative Proteomics with the *i2MassChroQ* Software Package. *Journal of Proteome Research* 23, 3353–3366. doi: 10.1021/acs.jproteome.3c00732

Former citation was :

Langella, O., Valot, B., Balliau, T., Blein-Nicolas, M., Bonhomme, L., and Zivy, M. (2017). X!TandemPipeline: A Tool to Manage Sequence Redundancy for Protein Inference and Phosphosite Identification. *Journal of Proteome Research* 16, 494–503. doi: 10.1021/acs.jproteome.6b00632

1.6 INSTALLATION OF THE SOFTWARE


The installation material is available at [HTTP://PAPPSO.INRAE.FR/EN/BIOINFO/XTANDEMPIPELINE/DOWNLOAD/](http://pappso.inrae.fr/en/bioinfo/xtandempipeline/download/).

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:

```
$ sudo apt install <package_name> 
```

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 
```

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

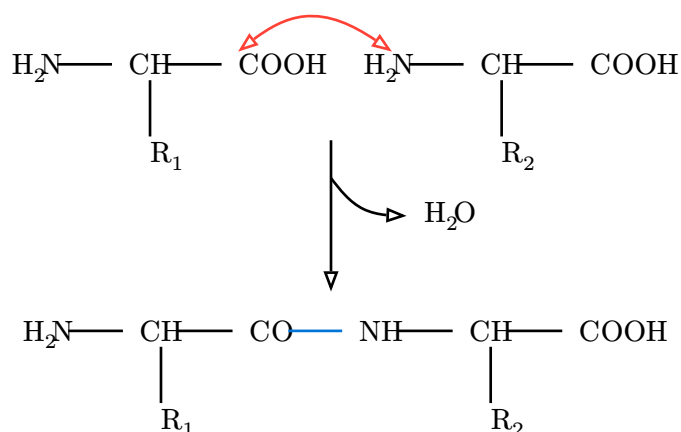
This section introduces the basics in protein polymer chemistry. The way this topic is going to be covered is admittedly biased towards mass spectrometry and proteins. Moreover, the aim of this chapter is to provide the reader with the specialized words that will later be used to describe and explain the (inner) workings of the *i2mcq* 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** 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 ($\text{H}_2\text{N}-$) 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**, 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).



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.

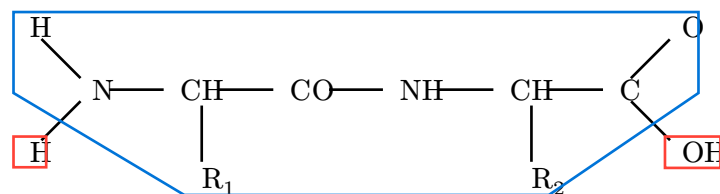
FIGURE 2.1: PEPTIDIC BOND FORMATION BY CONDENSATION



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, 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* (Rusconi, 2009), 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 spectrometrists. 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.

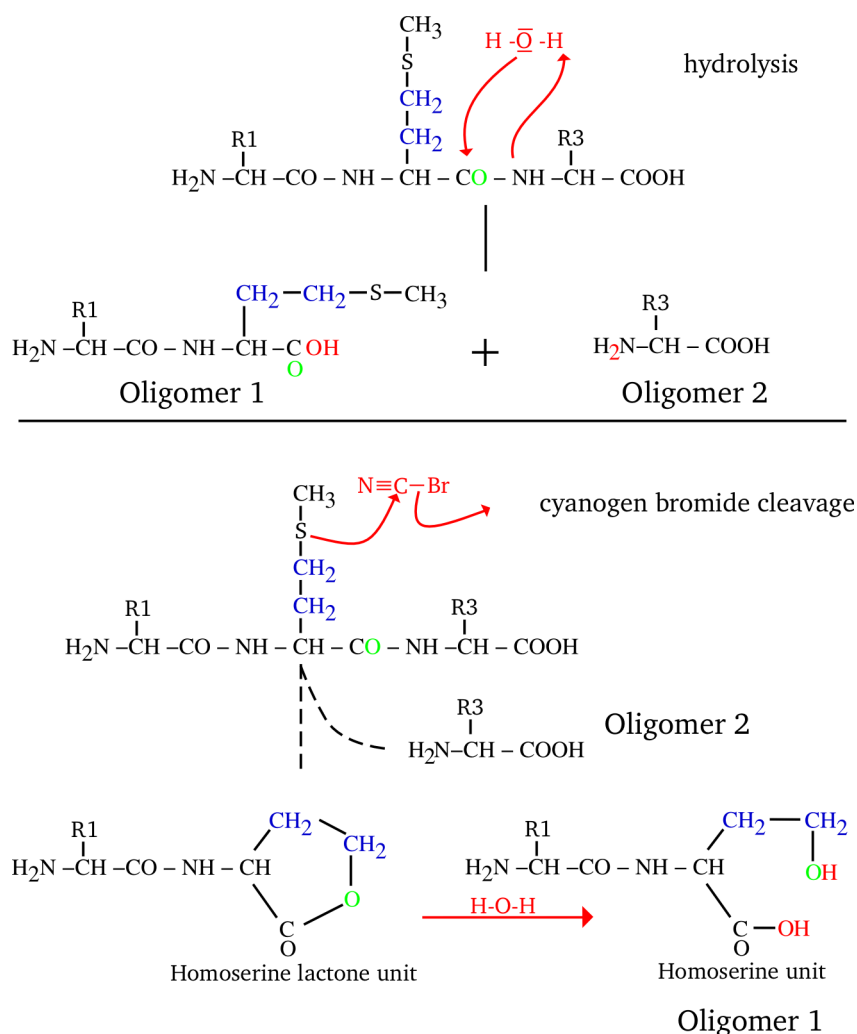
2.1.2.1 PROTEIN CLEAVAGE

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**, 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.

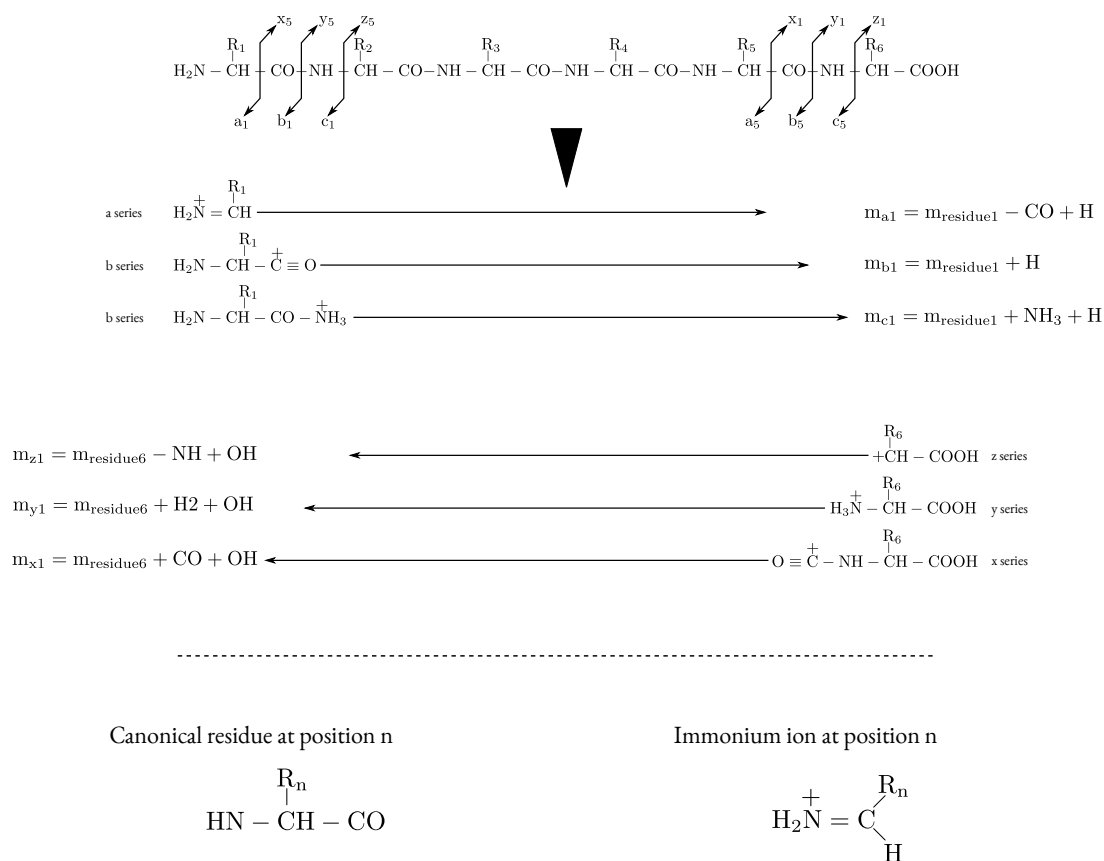
Another peculiarity of fragmentations, compared with cleavages, is the fact that there is no cleaving molecule starting the process, like water or cyanogen bromide, for example. Indeed, in the gas phase, the peptidic ions are “isolated”: that is, very far one from each other. A fragmentation process is often initiated by an intra molecular electron doublet rearrangement 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.



TIP

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**, 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 polymerization state).

By looking at **FIGURE 2.4**, 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* (a_3) would involve monomers $[1 \rightarrow]$ and the third fragment of series *y* (y_3) would involve monomers $[6 \rightarrow]$ (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 resolute 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;

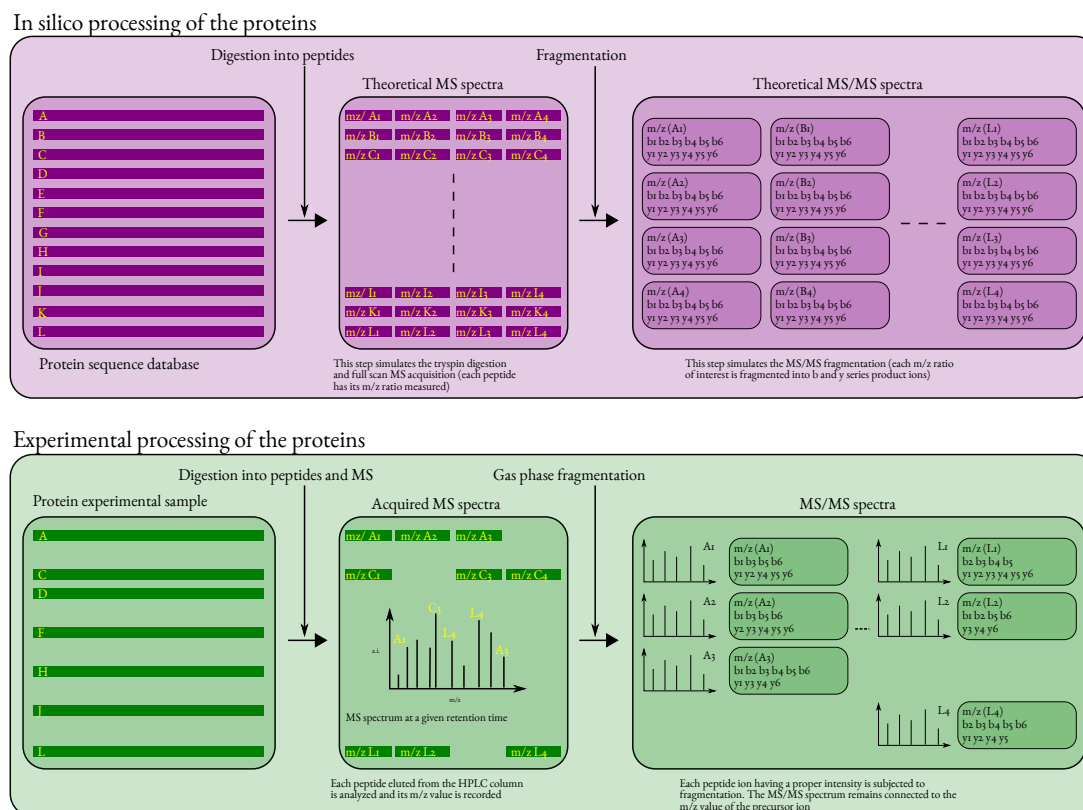
Bioinformatics The development and refinement of software that can cope with extremely large data sets (think metaproteomics) is one major field that enabled significant advances in proteomics. Also, refinement of algorithms related to the simulation of isotopic clusters and comparison with experimental 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**.



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 proteomics workflow is to digest all the proteins in the initial biological sample with a site-specific endoprotease: typically trypsin.

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 trypsinolysis 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 group³ 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.

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.

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 ($[\text{M}+\text{H}]^+$), while the ESI process mostly produces multi-charged ions ($[\text{M}+\text{nH}]^{n+}$). This has huge implications in the mass data analysis.

³If not either converted to an amide group by acetylation or formylation or cyclised.

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 fragmentation⁴.

These two kinds of data are necessary because the protein identification process is based on searches in protein databases using the precursor 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.

⁴Most 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 protein databases are files in the following FASTA format:

```
>GRMZM2G009506_P01 NP_001149383 serine/threonine-protein kinase receptor
MEEQH MAGPPYRYRLQHRRLMDIAPASASDDSGHHGSNGMAIMVSILVVIVCTLFYCV
YCWRWRKRNAVRRAQIERLRPMSSSDLPLMDLSSIHEATNSFSKENKLGEFGFPGVYRGV
MGGGAEIAVKRLSARSRQGAAEFRNEVELIAKLQHRNLVRLLGCCVERDEKMLVYEYLPN
RSLDSFLFDSRSGQLDWKTRQSIVLGIARGMLYLHEDSCLKVIHRDLKASNVLLDNRMN
PKISDFGMAKIFEEEGNEPNTGPVVGTYGYMAPEYAMEGVFSVKSDFVFSFGLVLEILSG
QRNGSMYLQEHQHTLIQDAWKLWNEDRAAEFMDAALAGSYPRDEAWRCFHVGLLCVQESP
DLRPTMSSVVLMLISDQTAQQMPAPAQPPLFASSRLGRKASASDLSLAMKTETTKQSVN
EVSISMMEPRFWADPGTNGAATSHPATGACKKRGQGGDRNVKDGLAARTPTHQPVARW
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:

- 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 &i2mcq; 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, top panel):

sequence The peptide's sequence;

m/z value The peptide's m/z value, often computed for the mono-protonated (&mh;) 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](#).

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](#).

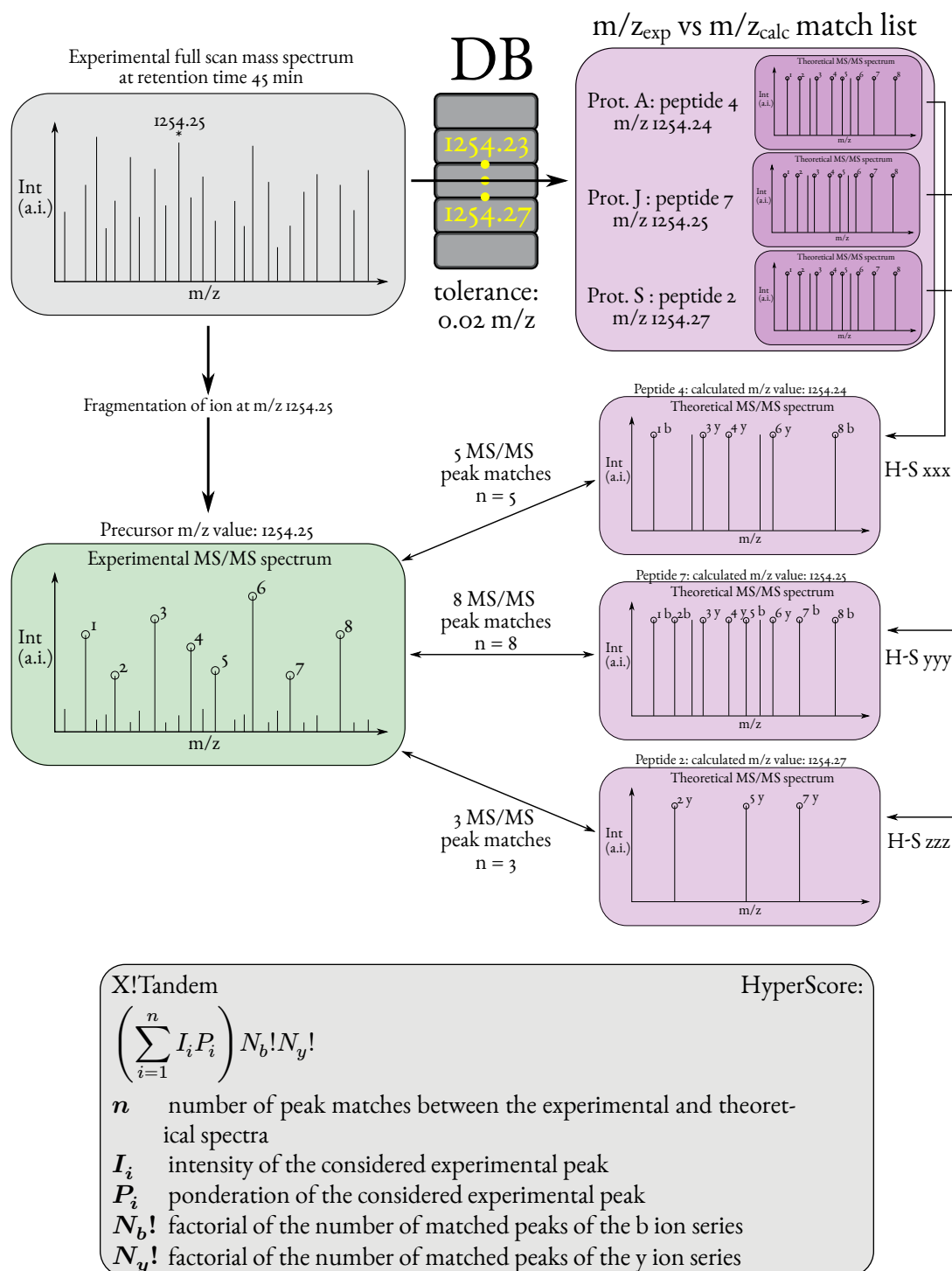
On the one hand (top panel, violet), the protein database is processed to digest *in silico* every protein it contains into a list of peptides. For each peptide arising from the digestion of a protein, the following data elements are recorded:

- 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 maintained.

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.



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](#)):

- 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](#)).

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 &xtandem;, 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](#)) 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](#), 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.

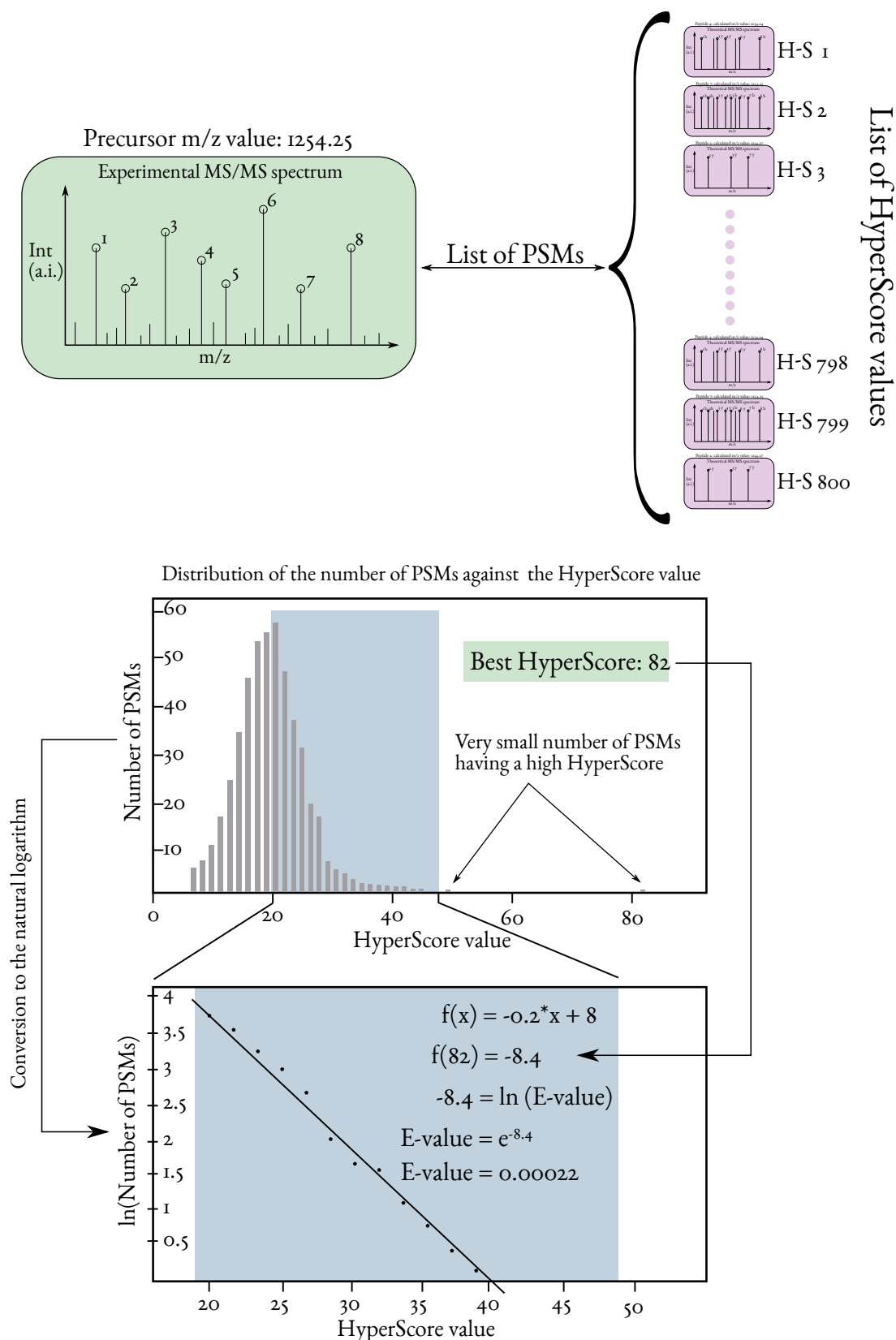
The best HyperScore value (82, in the example) is then used in the line equation to compute the corresponding ordinate (the natural logarithm of the PSMs count having that HyperScore). That value (-8.4, in the example) corresponds to the natural logarithm of the expectation value (E-value). By using 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.



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)



TIP

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).

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)

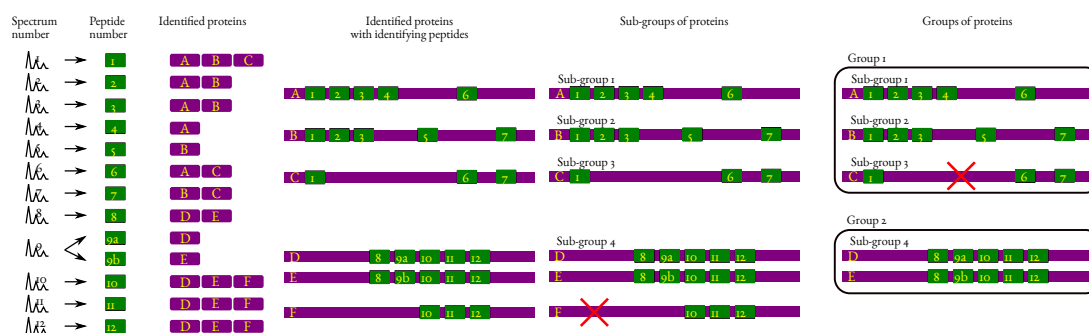
The last step in the computation of values that help the software and the user determine if identifications are faithful (for peptides and for proteins) is the computation of the protein expected value. This value is very easily computed: it is the product of the E-value of all the peptides that participated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

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 *parimony*. 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 (Langella et al., 2017). 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**, 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 protein identifications list occurs at two different steps in the inference process.



NOTE

The FDR is commonly computed as the ratio between the number of PSMs matching the decoy database over the number of PSMs matching the target database:

$$\text{FDR} = \frac{\# \text{decoy}}{\# \text{target}}$$

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 output 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 these steps:

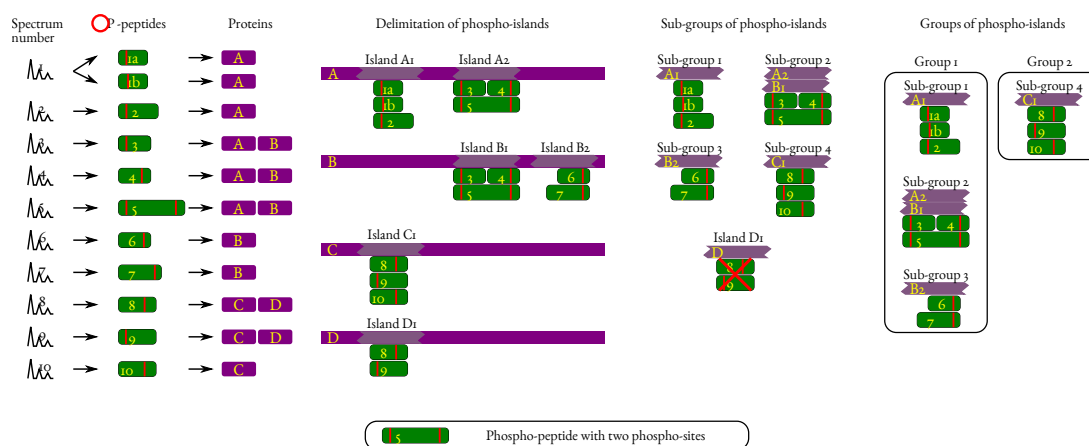
- Separate digestion of the samples (when there are more than one);
- Labeling of the peptides, each sample gets a different label;
- Pool of the whole set of peptides into a single mixture;
- Separation of the peptides on a strong cation exchange (SCX) resin, collection of the fractions;
- Phospho-peptide enrichment using IMAC⁵ 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 *i2mcq* is handling phospho-proteomics data. That algorithm is described below and in [FIGURE 2.9](#).

⁵Immobilized-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**, 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-peptides, 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

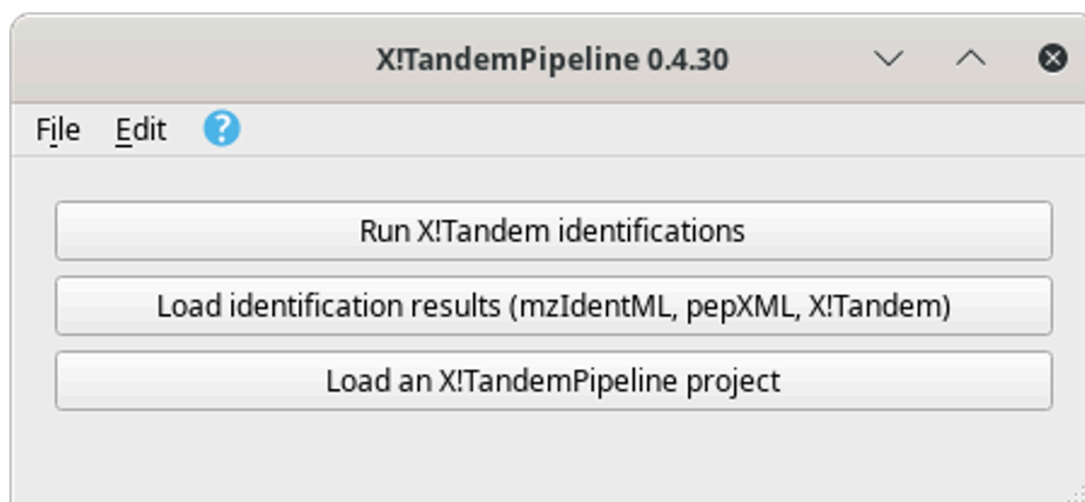
i2MassChroQ can also handle peptide vs spectrum matches data (peptide identification data) from other software with the following formats:

- 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](#), 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](#).



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 3.4.4 “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.



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*⁶.

Choose presets This setting defines the parameters that *&xtandem*; 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**.

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.



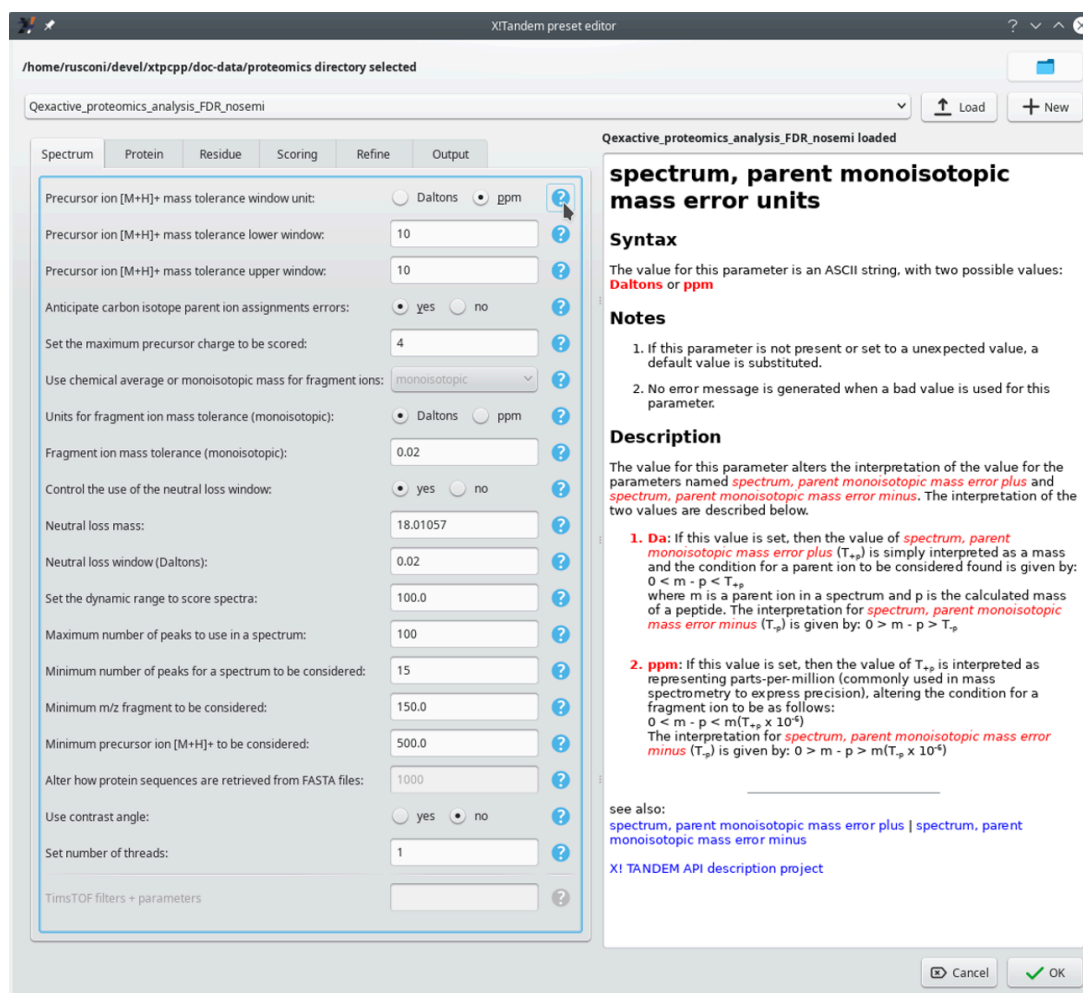
TIP

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

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**. Only the *Spectrum* tab is shown, but the interface is similar for all the other ones.

⁶See <https://research.cs.wisc.edu/htcondor/>.



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 page⁷ 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)

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). 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.

⁷The help texts are extracted automatically from the &xtandem; software documentation.

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

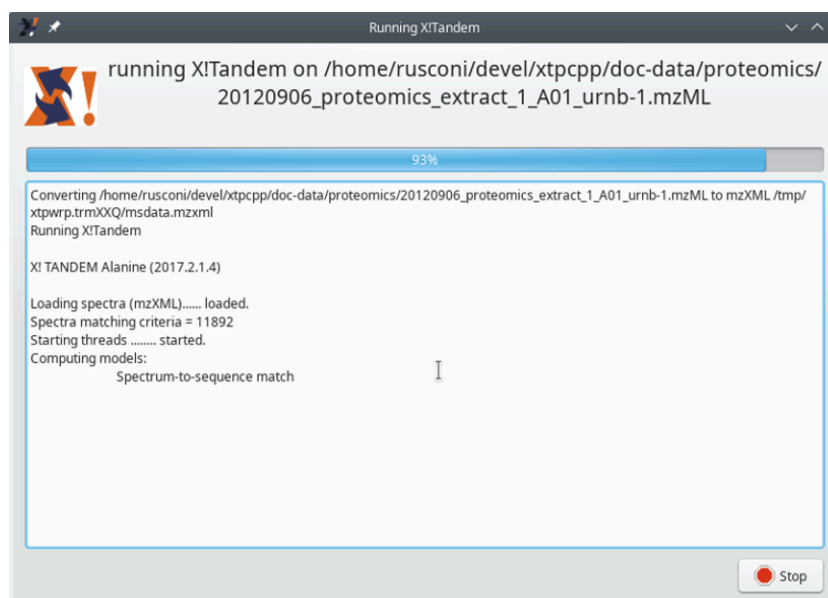
The dialog window pictured in [FIGURE 3.3](#) contains a number of tabs where various aspects of the *X!Tandem* run settings are handled. Each parameter's documentation can be seen on the pane on the right hand side of the window by clicking onto the question mark button next to it. These manual pages 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](#).

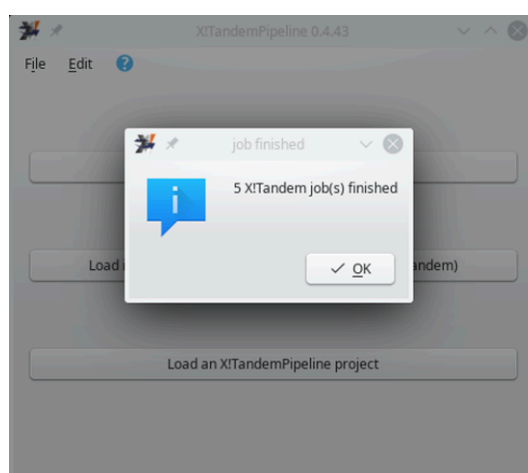
While the computation is carried over, the program shows the feedback dialog window pictured in [FIGURE 3.4](#).



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) albeit with a message shown in FIGURE 3.5.



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** and is described in the following section.

Load identification results

Results handling mode

☒ Combine
☐ Individual

Choose result files

/home/rusconi/devel/xtpcpp/doc-data/output.d/20120906_balliau_extract_1_A01_urnb-1.xml
/home/rusconi/devel/xtpcpp/doc-data/output.d/20120906_balliau_extract_1_A02_urzb-1.xml

Number of files: 2

Clear list

Add files

Contaminants

☒ Contaminants file
☐ Contaminant regular expression

contaminants_standards.fasta

Clear list

Add files

Contaminant removal mode

☐ Protein list
☒ Groups

Peptide and protein filters

Peptide threshold on:

☒ Evalue
☐ FDR

Peptide Evalue:

0.050000

Peptide FDR:

1.0%

Number of peptides per protein:

2

Overall samples:

☒

Protein Evalue:

0.01000000

Protein Evalue (log10):

-2.00

Pep repro:

1

Cancel

OK

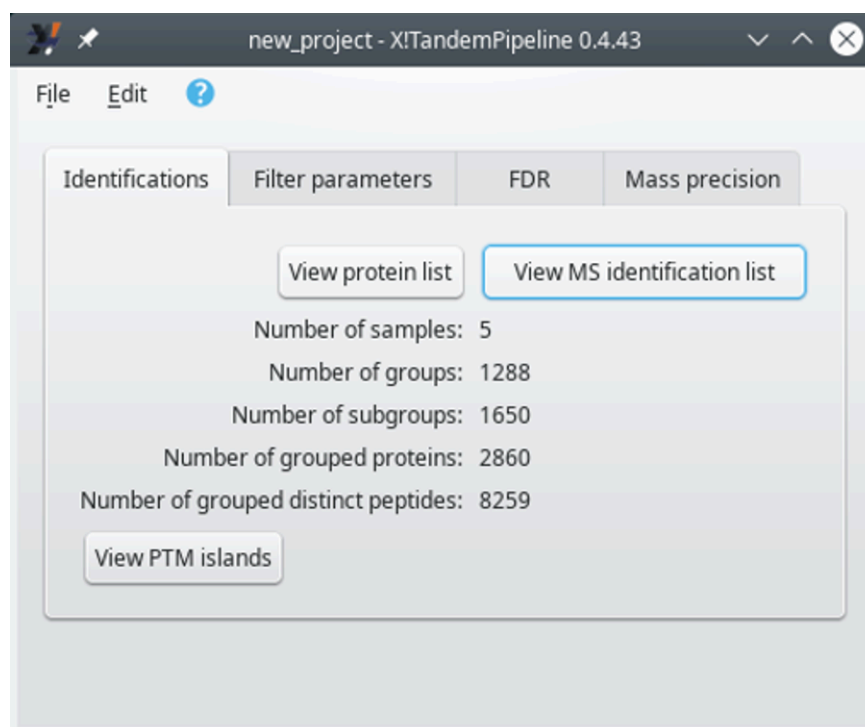
Loading identification results comes with some configuration that is described in the text.

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:

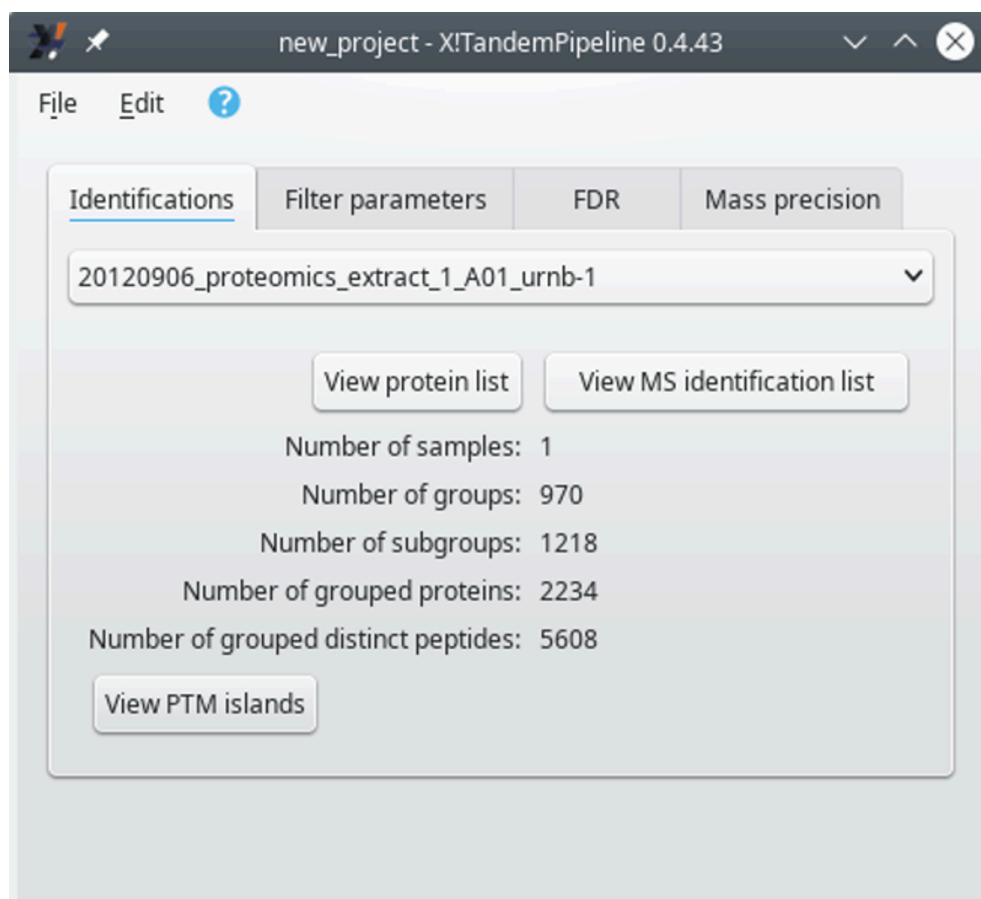
Combine in this mode, all the identification results coming from different identification results files are merged into a single set. That single set is the basis for the protein inference step and the identified proteins are listed into a single protein list window.



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 database 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](#)). When protein groups have sub-groups that contain a contaminant 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

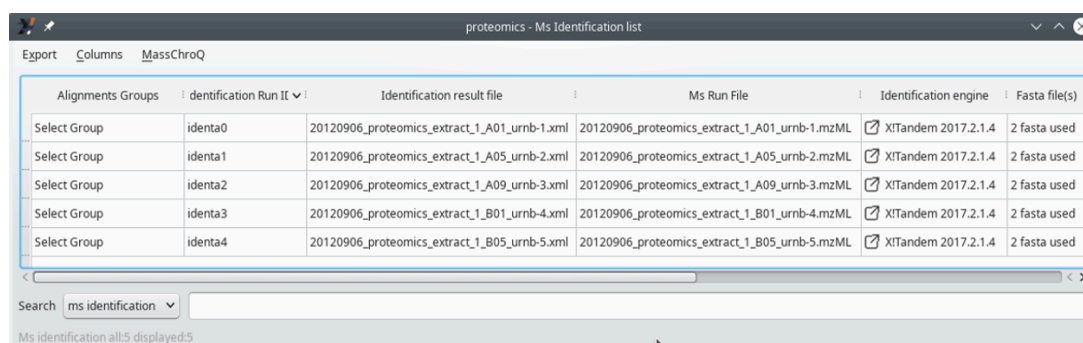
This setting needs to be checked in at least one case: when a complex peptidic mixture is separated by ion chromatography (typically on an SCX—strong cation exchange—resin) and the different fractions are analyzed by bottom-up proteomics. The peptides coming from a given protein might be located in different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

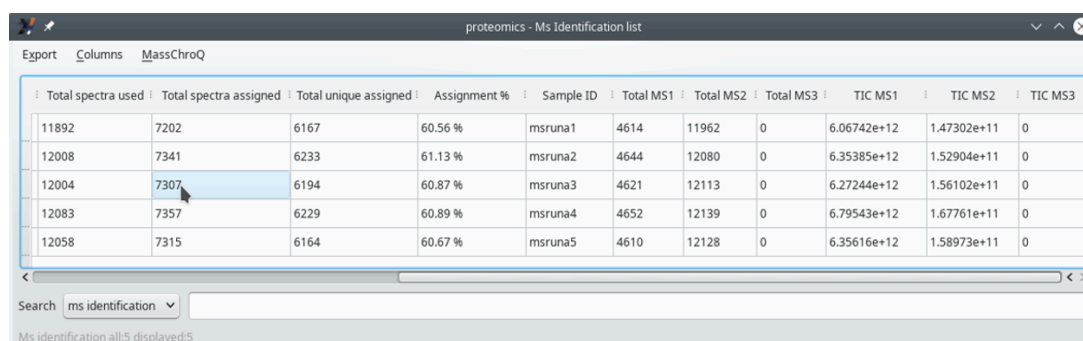
3.4.2 DISPLAYING THE MS IDENTIFICATIONS LIST



The screenshot shows a window titled 'proteomics - Ms Identification list'. It has a menu bar with 'Export', 'Columns', and 'MassChroQ'. Below the menu bar is a table with the following columns: 'Alignments Groups', 'identification Run ID', 'Identification result file', 'Ms Run File', 'Identification engine', and 'Fasta file(s)'. The table contains five rows of data, each starting with 'Select Group'. Below the table is a search bar with the text 'ms identification' and a dropdown arrow. At the bottom, it says 'Ms identification all:5 displayed:5'.

Alignments Groups	identification Run ID	Identification result file	Ms Run File	Identification engine	Fasta file(s)
Select Group	identa0	20120906_proteomics_extract_1_A01_urnb-1.xml	20120906_proteomics_extract_1_A01_urnb-1.mzML	X!Tandem 2017.2.1.4	2 fasta used
Select Group	identa1	20120906_proteomics_extract_1_A05_urnb-2.xml	20120906_proteomics_extract_1_A05_urnb-2.mzML	X!Tandem 2017.2.1.4	2 fasta used
Select Group	identa2	20120906_proteomics_extract_1_A09_urnb-3.xml	20120906_proteomics_extract_1_A09_urnb-3.mzML	X!Tandem 2017.2.1.4	2 fasta used
Select Group	identa3	20120906_proteomics_extract_1_B01_urnb-4.xml	20120906_proteomics_extract_1_B01_urnb-4.mzML	X!Tandem 2017.2.1.4	2 fasta used
Select Group	identa4	20120906_proteomics_extract_1_B05_urnb-5.xml	20120906_proteomics_extract_1_B05_urnb-5.mzML	X!Tandem 2017.2.1.4	2 fasta used

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)



The screenshot shows the same window as Figure 3.9, but with different columns displayed. The columns are: 'Total spectra used', 'Total spectra assigned', 'Total unique assigned', 'Assignment %', 'Sample ID', 'Total MS1', 'Total MS2', 'Total MS3', 'TIC MS1', 'TIC MS2', and 'TIC MS3'. The table contains five rows of data, each starting with a sample ID. The row for 'msruna3' is highlighted. Below the table is a search bar with the text 'ms identification' and a dropdown arrow. At the bottom, it says 'Ms identification all:5 displayed:5'.

Total spectra used	Total spectra assigned	Total unique assigned	Assignment %	Sample ID	Total MS1	Total MS2	Total MS3	TIC MS1	TIC MS2	TIC MS3
11892	7202	6167	60.56 %	msruna1	4614	11962	0	6.06742e+12	1.47302e+11	0
12008	7341	6233	61.13 %	msruna2	4644	12080	0	6.35385e+12	1.52904e+11	0
12004	7307	6194	60.87 %	msruna3	4621	12113	0	6.27244e+12	1.56102e+11	0
12083	7357	6229	60.89 %	msruna4	4652	12139	0	6.79543e+12	1.67761e+11	0
12058	7315	6164	60.67 %	msruna5	4610	12128	0	6.35616e+12	1.58973e+11	0

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 modification 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 3.4.4 “LOADING *i2MASSCHROQ* PROJECTS”](#) for loading such a project.

3.4.4 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](#).

checked	group	accession	description	log(Evalue)	Evalue	spectra	specific spectra	sequences	specific sequence	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	B65IF9 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	GRMZM2G049496_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	GRMZM2G045611_P01	B4FBL7 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
✓		GRMZM2G099232_P01	NP_001141257 hypothetical ...	-2.5376	0.0029	2		1		2.49 %	58693.1	0.0416667	0.100694
✓		GRMZM2G099232_P02	NP_001141257 hypothetical ...	-2.5376	0.0029	2		1		3.95 %	37105.2	0.0769231	0.193777
✓		GRMZM2G099232_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	0.370339
✓	c330.a1.a1	GRMZM2G157462_P01	NP_001152484 dynamin-2A ...	-20.5331	2.93026e-21	7	7	4	4	4.93 %	95589.6	0.0851064	0.216484

The protein identifications list window displays the proteins assembled into groups. A number of metadata about the identifications are shown in a number of columns, 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/](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/](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 fulfill 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.

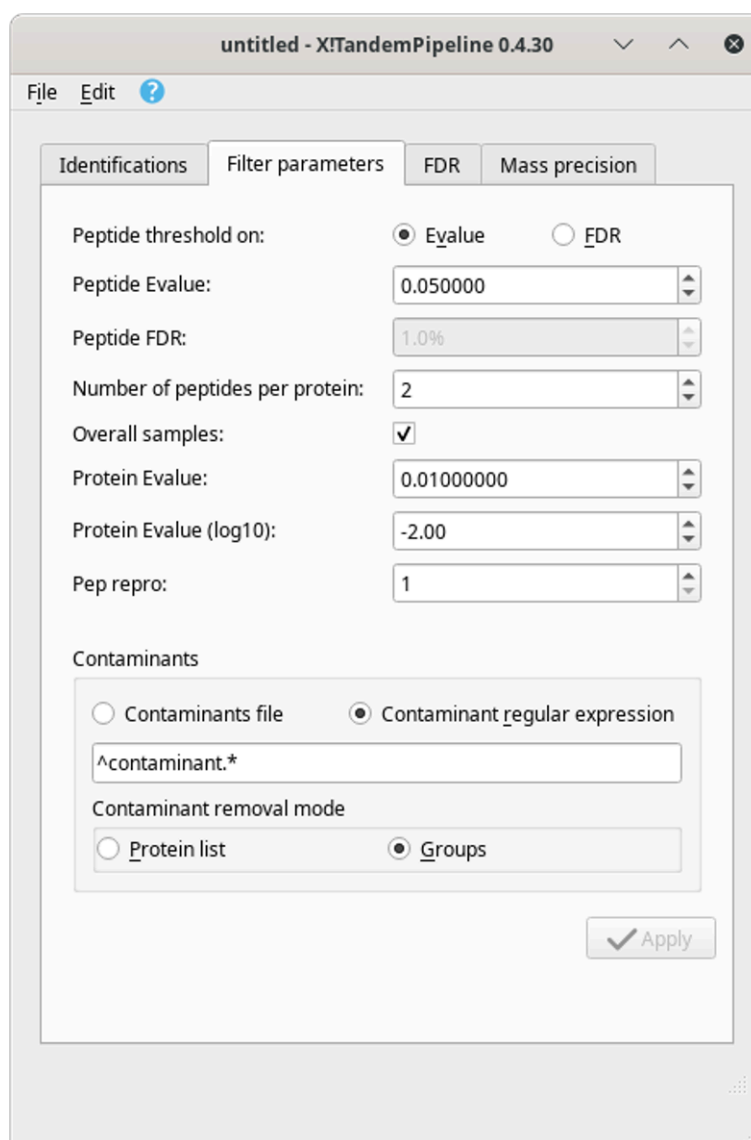
The protein identifications list table view above shows greyed protein identities. These are proteins that, by current filter parameters (E-value threshold, for example), are considered *not* valid.

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.

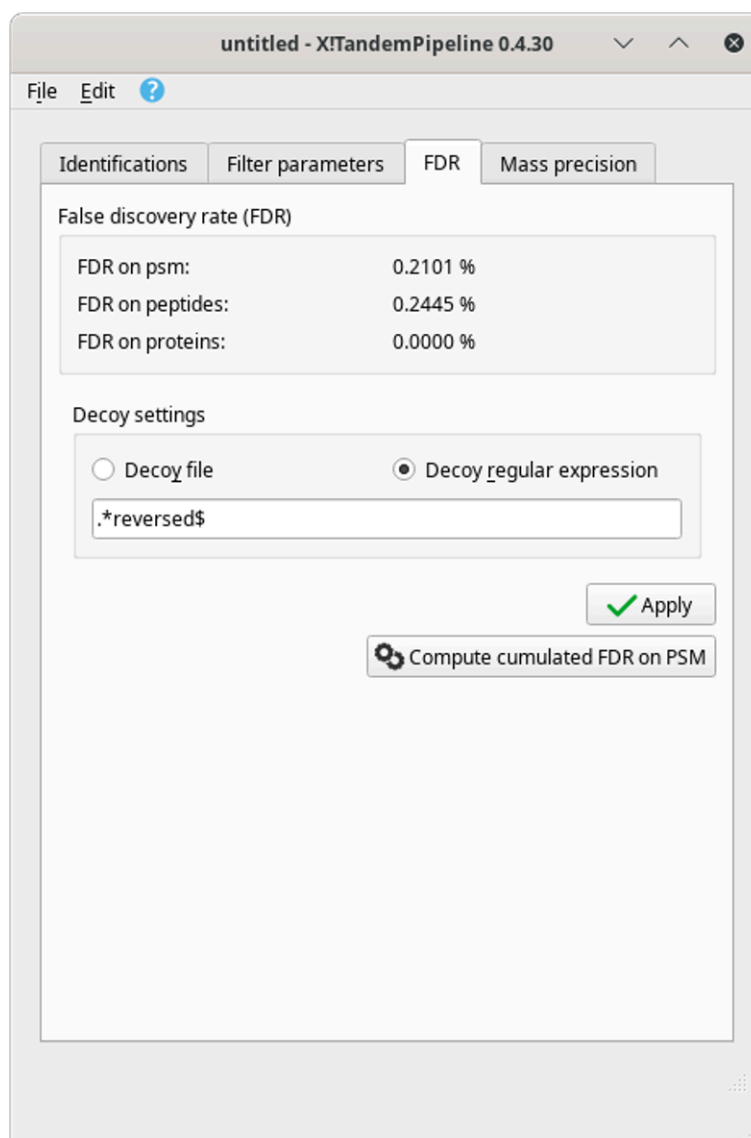
Dynamic setting of the filter parameters. *i2MassChroQ* provides a rather high level of flexibility: once a protein identification results set of files has been loaded and that the protein inference process is achieved, the resulting protein groups are displayed in the *Protein list* window. At this 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](#).



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

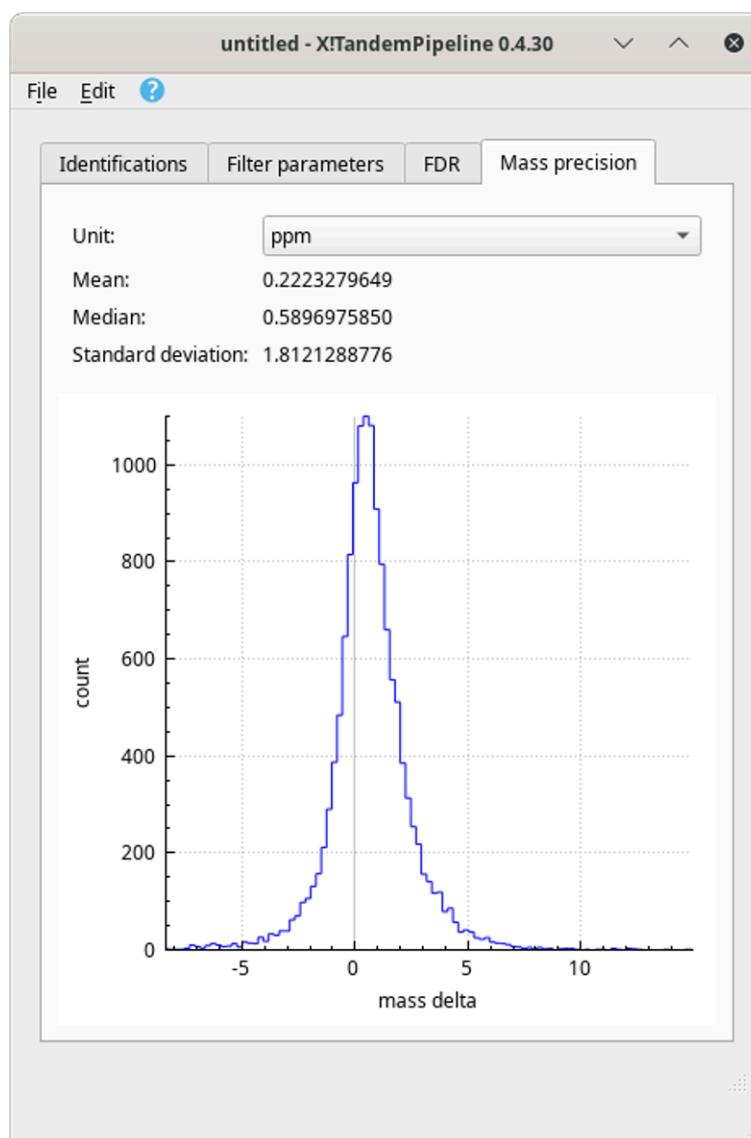
Real time update of the false discovery rate. The false discovery rate (FDR) is recalculated at each protein inference process. The data regarding this quality assessment criterion are shown in [FIGURE 4.3](#).



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](#). 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.



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 identified 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.

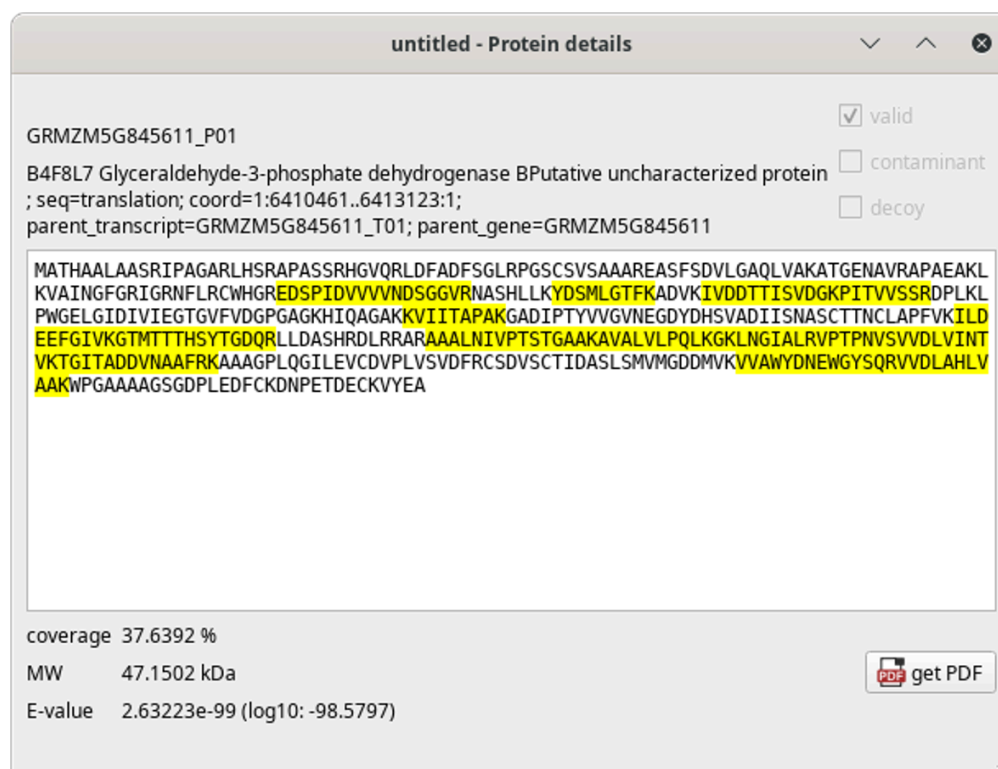
4.1.3 DELVING INSIDE THE PROTEIN IDENTIFICATION DATA

The protein list table view, as pictured in **FIGURE 4.1** 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.

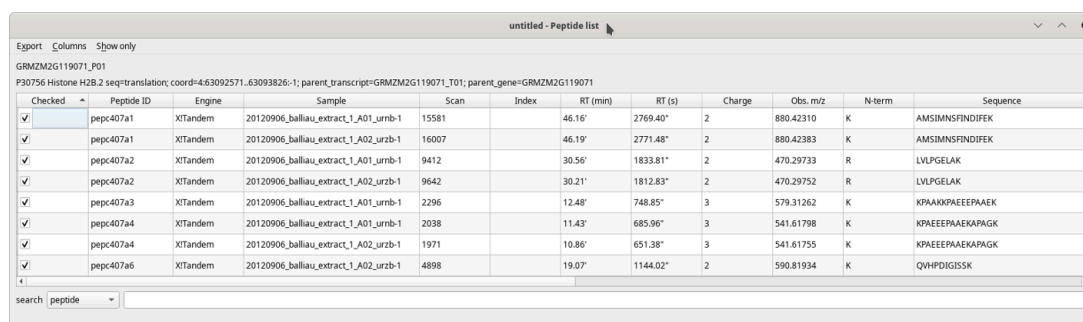
It is possible to have multiple windows opened at a time by clicking a new row while maintaining the `Ctrl` key pressed.

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.

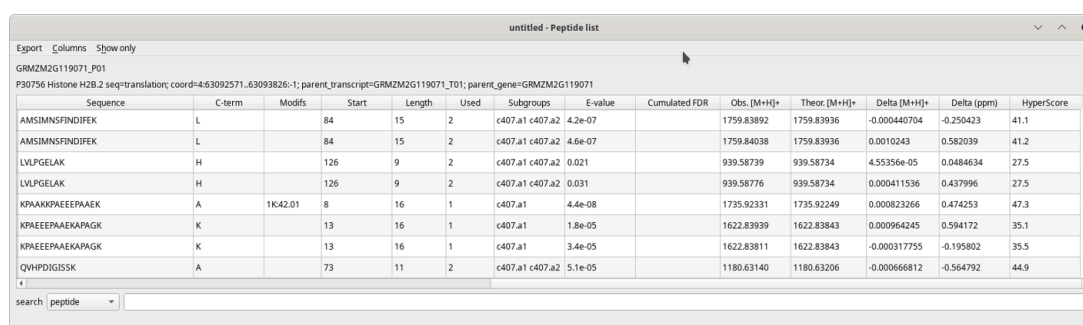


The screenshot shows the 'untitled - Peptide list' window. The table has the following columns: Checked, Peptide ID, Engine, Sample, Scan, Index, RT (min), RT (s), Charge, Obs. m/z, N-term, and Sequence. The data rows are as follows:

Checked	Peptide ID	Engine	Sample	Scan	Index	RT (min)	RT (s)	Charge	Obs. m/z	N-term	Sequence
<input checked="" type="checkbox"/>	pepc407a1	XITandem	20120906_balliau_extract_1_A01_urmb-1	15581		46.16*	2769.40*	2	880.42310	K	AMSIMNSFINDIFEK
<input checked="" type="checkbox"/>	pepc407a1	XITandem	20120906_balliau_extract_1_A02_urzb-1	16007		46.19*	2771.48*	2	880.42383	K	AMSIMNSFINDIFEK
<input checked="" type="checkbox"/>	pepc407a2	XITandem	20120906_balliau_extract_1_A01_urmb-1	9412		30.56*	1833.81*	2	470.29733	R	LVLPGELAK
<input checked="" type="checkbox"/>	pepc407a2	XITandem	20120906_balliau_extract_1_A02_urzb-1	9642		30.21*	1812.83*	2	470.29752	R	LVLPGELAK
<input checked="" type="checkbox"/>	pepc407a3	XITandem	20120906_balliau_extract_1_A01_urmb-1	2296		12.48*	748.85*	3	579.31262	K	KPAAKKPAEEEPAAEK
<input checked="" type="checkbox"/>	pepc407a4	XITandem	20120906_balliau_extract_1_A01_urmb-1	2038		11.43*	685.96*	3	541.61798	K	KPAEEEPAAEKAPAGK
<input checked="" type="checkbox"/>	pepc407a4	XITandem	20120906_balliau_extract_1_A02_urzb-1	1971		10.86*	651.38*	3	541.61755	K	KPAEEEPAAEKAPAGK
<input checked="" type="checkbox"/>	pepc407a6	XITandem	20120906_balliau_extract_1_A02_urzb-1	4898		19.07*	1144.02*	2	590.81934	K	QVHPDIGISSK

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

FIGURE 4.6: THE PEPTIDE LIST WINDOW (FIRST COLUMNS)



The screenshot shows the 'untitled - Peptide list' window. The table has the following columns: Sequence, C-term, Modifs, Start, Length, Used, Subgroups, E-value, Cumulated FDR, Obs. [M+H]⁺, Theor. [M+H]⁺, Delta [M+H]⁺, Delta (ppm), and HyperScore. The data rows are as follows:

Sequence	C-term	Modifs	Start	Length	Used	Subgroups	E-value	Cumulated FDR	Obs. [M+H] ⁺	Theor. [M+H] ⁺	Delta [M+H] ⁺	Delta (ppm)	HyperScore
AMSIMNSFINDIFEK	L		84	15	2	c407.a1 c407.a2	4.2e-07		1759.83892	1759.83936	-0.000440704	-0.250423	41.1
AMSIMNSFINDIFEK	L		84	15	2	c407.a1 c407.a2	4.6e-07		1759.84038	1759.83936	0.0010243	0.582039	41.2
LVLPGELAK	H		126	9	2	c407.a1 c407.a2	0.021		939.58739	939.58734	4.55356e-05	0.0484634	27.5
LVLPGELAK	H		126	9	2	c407.a1 c407.a2	0.031		939.58776	939.58734	0.000411536	0.437996	27.5
KPAAKKPAEEEPAAEK	A	1K42.01	8	16	1	c407.a1	4.4e-08		1735.92331	1735.92249	0.000823266	0.474253	47.3
KPAEEEPAAEKAPAGK	K		13	16	1	c407.a1	1.8e-05		1622.83939	1622.83843	0.000964245	0.594172	35.1
KPAEEEPAAEKAPAGK	K		13	16	1	c407.a1	3.4e-05		1622.83811	1622.83843	-0.000317755	-0.195802	35.5
QVHPDIGISSK	A		73	11	2	c407.a1 c407.a2	5.1e-05		1180.63140	1180.63206	-0.000666812	-0.564792	44.9

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](#)

<input checked="" type="checkbox"/> Checked	<input checked="" type="checkbox"/> Inter prophet prob.
<input checked="" type="checkbox"/> Peptide ID	<input checked="" type="checkbox"/> HyperScore
<input checked="" type="checkbox"/> Engine	<input checked="" type="checkbox"/> Mascot score
<input checked="" type="checkbox"/> Sample	<input checked="" type="checkbox"/> Mascot E-value
<input checked="" type="checkbox"/> Scan	<input checked="" type="checkbox"/> OMSSA E-value
<input checked="" type="checkbox"/> Index	<input checked="" type="checkbox"/> OMSSA p-value
<input checked="" type="checkbox"/> RT (min)	<input checked="" type="checkbox"/> MS-GF raw score
<input checked="" type="checkbox"/> RT (s)	<input checked="" type="checkbox"/> MS-GF de novo
<input checked="" type="checkbox"/> Charge	<input checked="" type="checkbox"/> MS-GF energy
<input checked="" type="checkbox"/> Obs. m/z	<input checked="" type="checkbox"/> MS-GF spectral E-value
<input checked="" type="checkbox"/> N-term	<input checked="" type="checkbox"/> MS-GF E-value
<input checked="" type="checkbox"/> Sequence	<input checked="" type="checkbox"/> MS-GF isotope error
<input checked="" type="checkbox"/> C-term	<input checked="" type="checkbox"/> Comet XCorr
<input checked="" type="checkbox"/> Modifs	<input checked="" type="checkbox"/> Comet DeltaCn
<input checked="" type="checkbox"/> Label	<input checked="" type="checkbox"/> Comet DeltaCnStar
<input checked="" type="checkbox"/> Start	<input checked="" type="checkbox"/> Comet SpScore
<input checked="" type="checkbox"/> Length	<input checked="" type="checkbox"/> Comet SpRank
<input checked="" type="checkbox"/> Used	<input checked="" type="checkbox"/> Comet E-value
<input checked="" type="checkbox"/> Subgroups	<input checked="" type="checkbox"/> DeepProt matched peaks
<input checked="" type="checkbox"/> E-value	<input checked="" type="checkbox"/> DeepProt fitted peaks
<input checked="" type="checkbox"/> Cumulated FDR	<input checked="" type="checkbox"/> DeepProt match type
<input checked="" type="checkbox"/> Obs. [M+H] ⁺	<input checked="" type="checkbox"/> DeepProt status
<input checked="" type="checkbox"/> Theor. [M+H] ⁺	<input checked="" type="checkbox"/> DeepProt mass delta
<input checked="" type="checkbox"/> Delta [M+H] ⁺	<input checked="" type="checkbox"/> DeepProt mass delta pos.
<input checked="" type="checkbox"/> Delta (ppm)	
<input checked="" type="checkbox"/> Prophet prob.	

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.

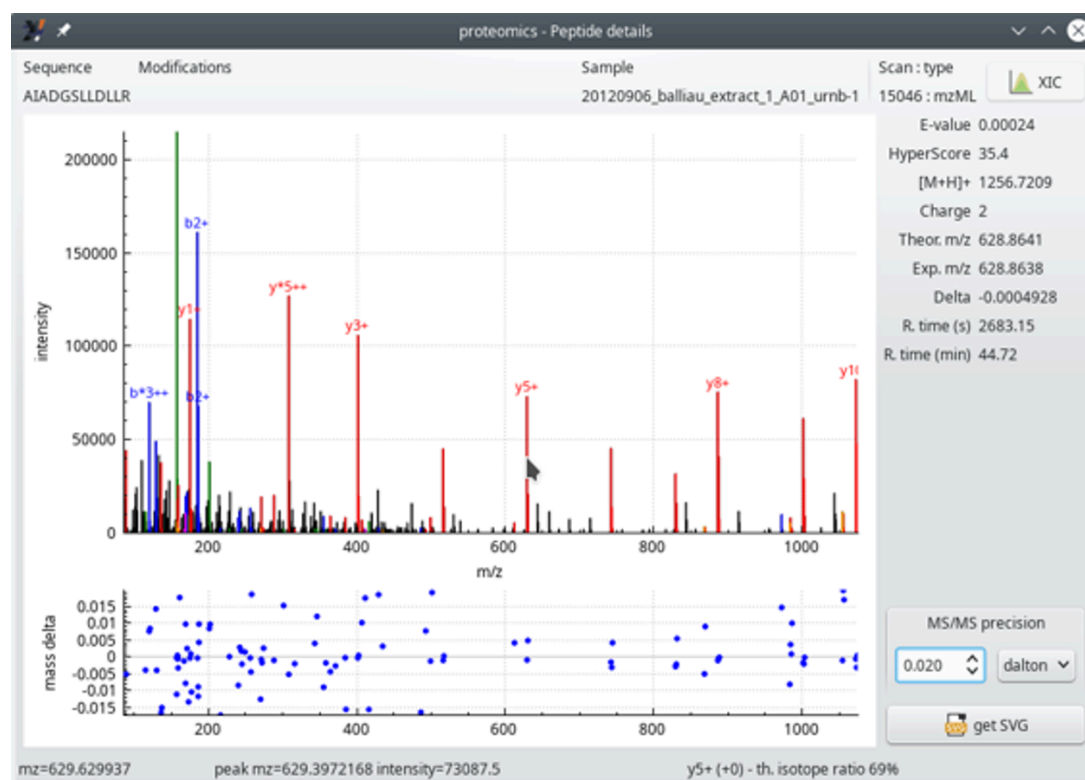
Exporting the final protein identifications list to a spread sheet. Once all the peptides in the identifications list have been properly checked, the user might export the data set to an OpenDocumentFormat (ODF) spread sheet file using the *As ODS file* menu item of the main window's *Export* menu.

4.2.3 DELVING INSIDE THE PEPTIDE IDENTIFICATION DATA

The *Peptide list* table view, as pictured in FIGURE 4.6 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](#).



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](#), 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 assignment 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.

BIBLIOGRAPHY

- Langella, O., Renne, T., Balliau, T., Davanture, M., Brehmer, S., Zivy, M., et al. (2024). Full Native timsTOF PASEF-Enabled Quantitative Proteomics with the i2MassChroQ Software Package. *Journal of Proteome Research* 23, 3353–3366. doi: 10.1021/acs.jproteome.3c00732
- Langella, O., Valot, B., Balliau, T., Blein-Nicolas, M., Bonhomme, L., and Zivy, M. (2017). X!TandemPipeline: A Tool to Manage Sequence Redundancy for Protein Inference and Phosphosite Identification. *Journal of Proteome Research* 16, 494–503. doi: 10.1021/acs.jproteome.6b00632
- Rusconi, F. (2009). massXpert 2 : a cross-platform software environment for polymer chemistry modelling and simulation/analysis of mass spectrometric data. *Bioinformatics* 25, 2741–2742. doi: 10.1093/bioinformatics/btp504