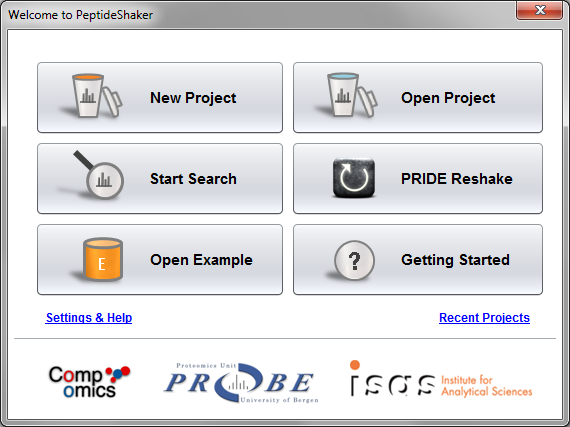
1. PSM, Peptide & Protein Visualization

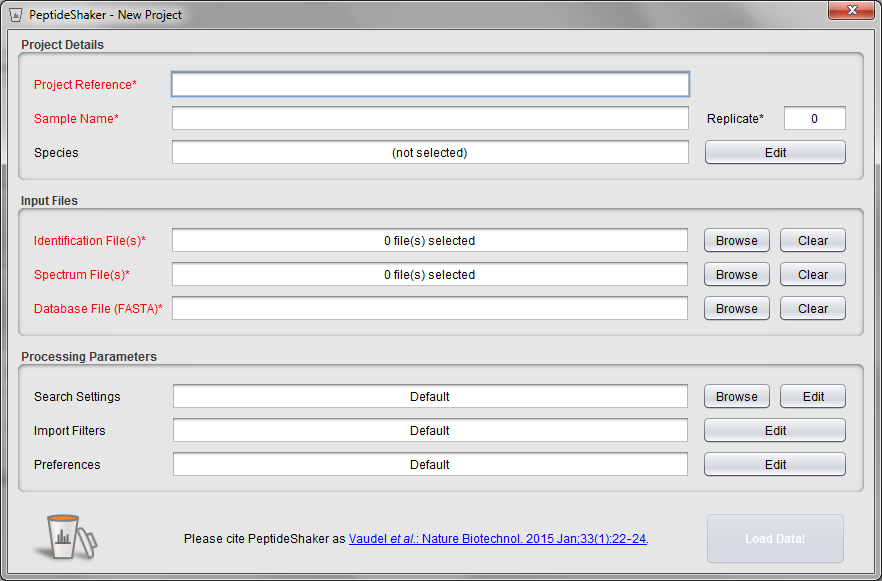
The search conducted in the Peptide to Spectrum Matching chapter generated two files containing the peptides matched by OMSSA and X!Tandem for each spectrum, so-called Peptide to Spectrum Matches (PSMs). From these we want to find the identified peptides and proteins. This is the task of PeptideShaker1 ([http://peptide-shaker.googlecode.com](http://peptide-shaker.googlecode.com/)).

First, start PeptideShaker by double-clicking the file called PeptideShaker-X.Y.Z.jar in the PeptideShaker-X.Y.Z folder located in the software folder (X.Y.Z is the version number). You should see the following dialog:



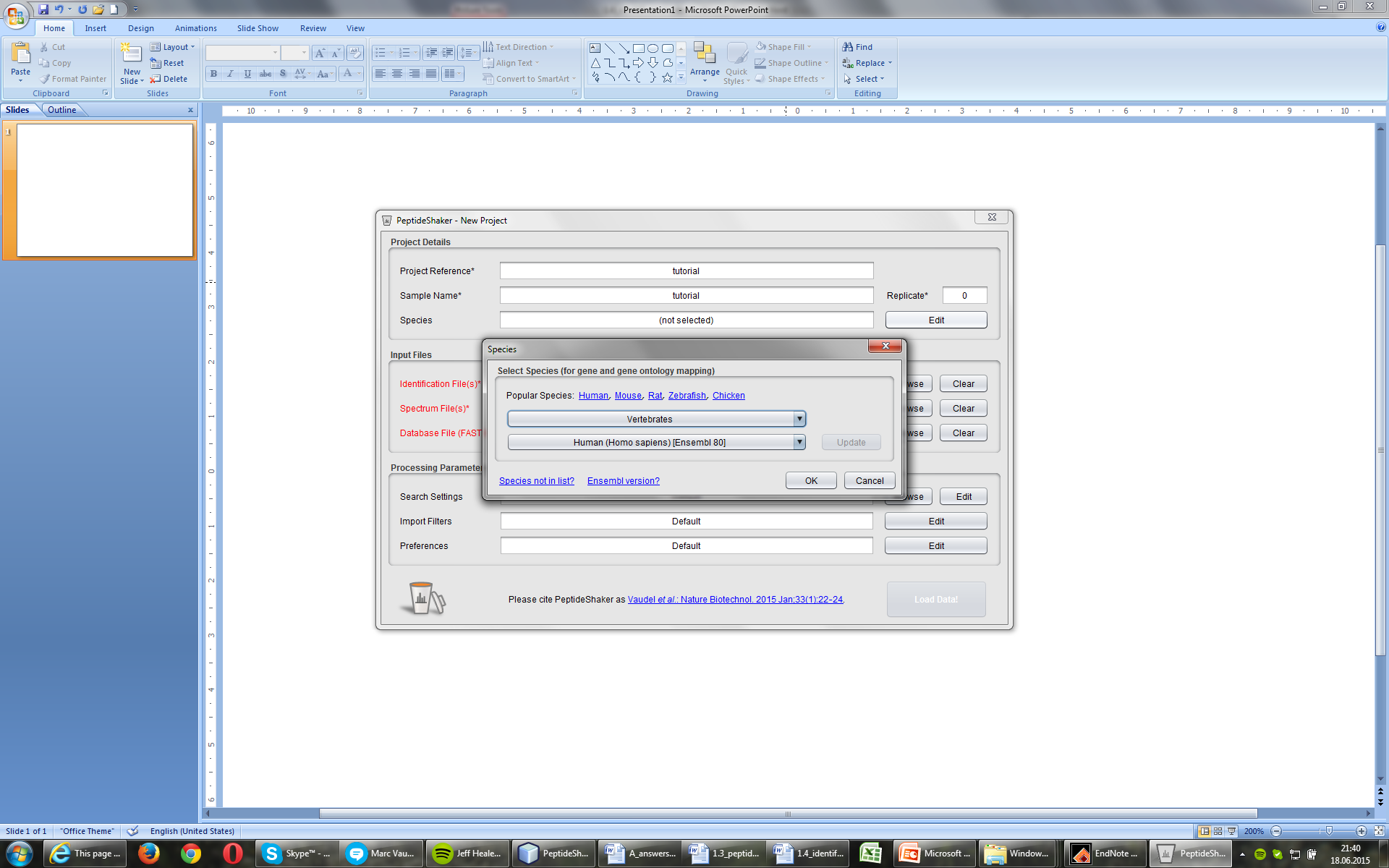
From the Welcome dialog, you can create a new project, open a previously saved project, start a search with SearchGUI, Reshake (i.e., reanalyze) existing PRIDE[2](#_ENREF_1) data, open an example dataset or navigate our ‘Getting Started' presentation.

We are now going to create a new project with the previously generated OMSSA and X!Tandem files (also available in the resources folder). Click on ‘New Project’. You will see this screen:



Start by giving your project and sample a name at the top, and leave replicate number at 0. We will not take advantage of these details in this exercise, but such annotation is crucial for later (re-)use for real projects!

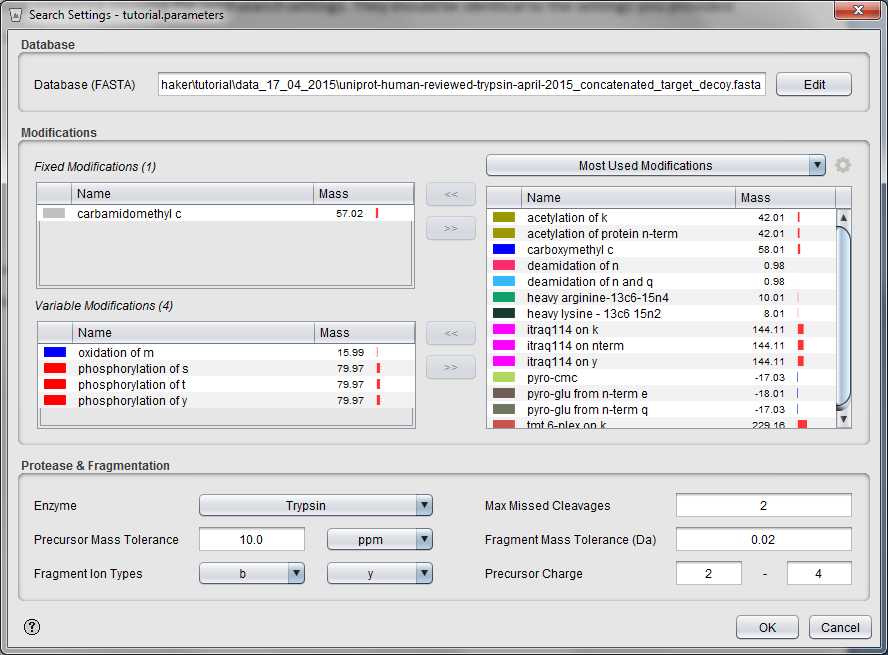
The species of interest for this sample is human: this is in fact a measurement of a HeLa lysate, courtesy of the Leibniz-Institut für Analytische Wissenschaften - ISAS - e. V., Dortmund, Germany. Click the ‘Edit’ button for ‘Species‘ and select ‘Human (Homo Sapiens) [Ensembl 80]’ as displayed below. *What is Ensembl? [1.4a]*



Now we are ready to add the search result files. Click the ‘Browse’ button next to the ‘Identification File(s)’ text field. Navigate to the resources folder, select the zip file containing the OMSSA and X!Tandem result files (searchgui\_out.zip) and click ‘Add’.

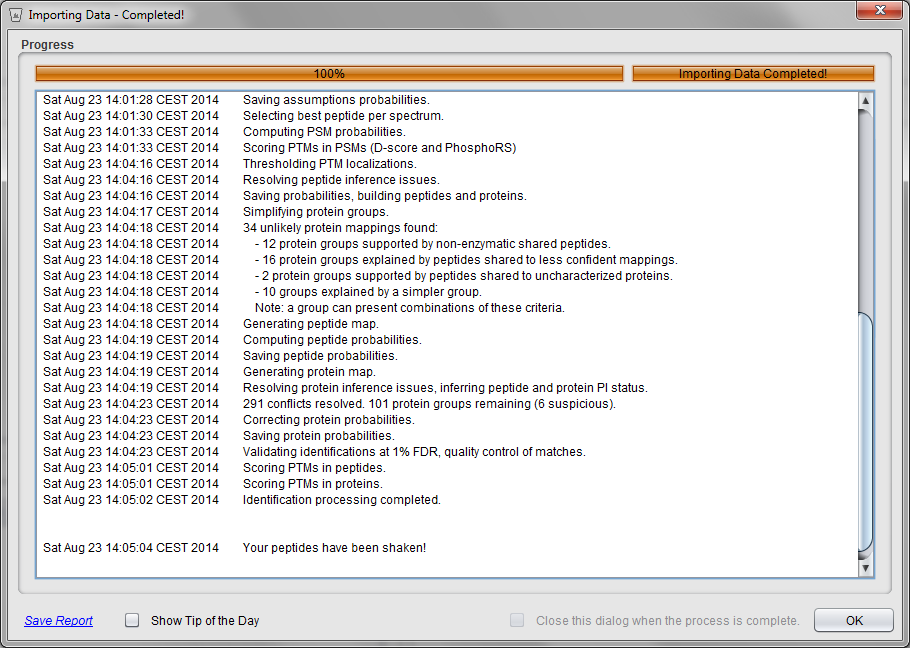
Note that the spectrum files and the database are automatically filled in when loading results from SearchGUI. Otherwise you have to load these manually (files are in the resources folder). Note also that the descriptions of the processing parameters have changed. This information was extracted from the searchGUI\_input.txt file and the parameters file created by SearchGUIalong with the identification files. Note that if files have been moved in the interim, you can add them manually.

Once all the files have been loaded, click the ‘Edit’ button for the ‘Search Settings‘. You should see this dialog appearing showing the used search settings. They should be identical to the settings you provided in SearchGUI:



In addition to the modifications listed here you may also see the following modifications in your search results: acetylation, pyro-cmc and pyro-glu. *Where do these come from? [1.4b]*

Click ‘OK’ to go back to the New Project dialog and then click ‘Load Data!’. The selected files will now be processed and loaded into PeptideShaker, and you can follow the progress in the dialog while browsing some PeptideShaker usage tips.



**Tip:**  
*If you want to save time you can cancel the loading and open the PeptideShaker example dataset instead. This was created using the exact same input as above and is available in the Welcome Dialog.*

Loading large dataset in PeptideShaker can be time consuming. The processing will be faster if increasing the memory allocated to PeptideShaker. You can do that via the *Edit > Java Settings* menu (or *via* the Welcome Dialog - *Settings & Help > Settings > Java Settings*). Note that large datasets (>100 000 spectra) or databases (>100 000 protein sequences) will require a lot of memory. For smaller datasets (<100 000 spectra and <100 000 protein sequences), a standard laptop (64 bits and 4 GB of RAM) is sufficient. The performance will also be improved if working with SSD discs and multiple CPUs.

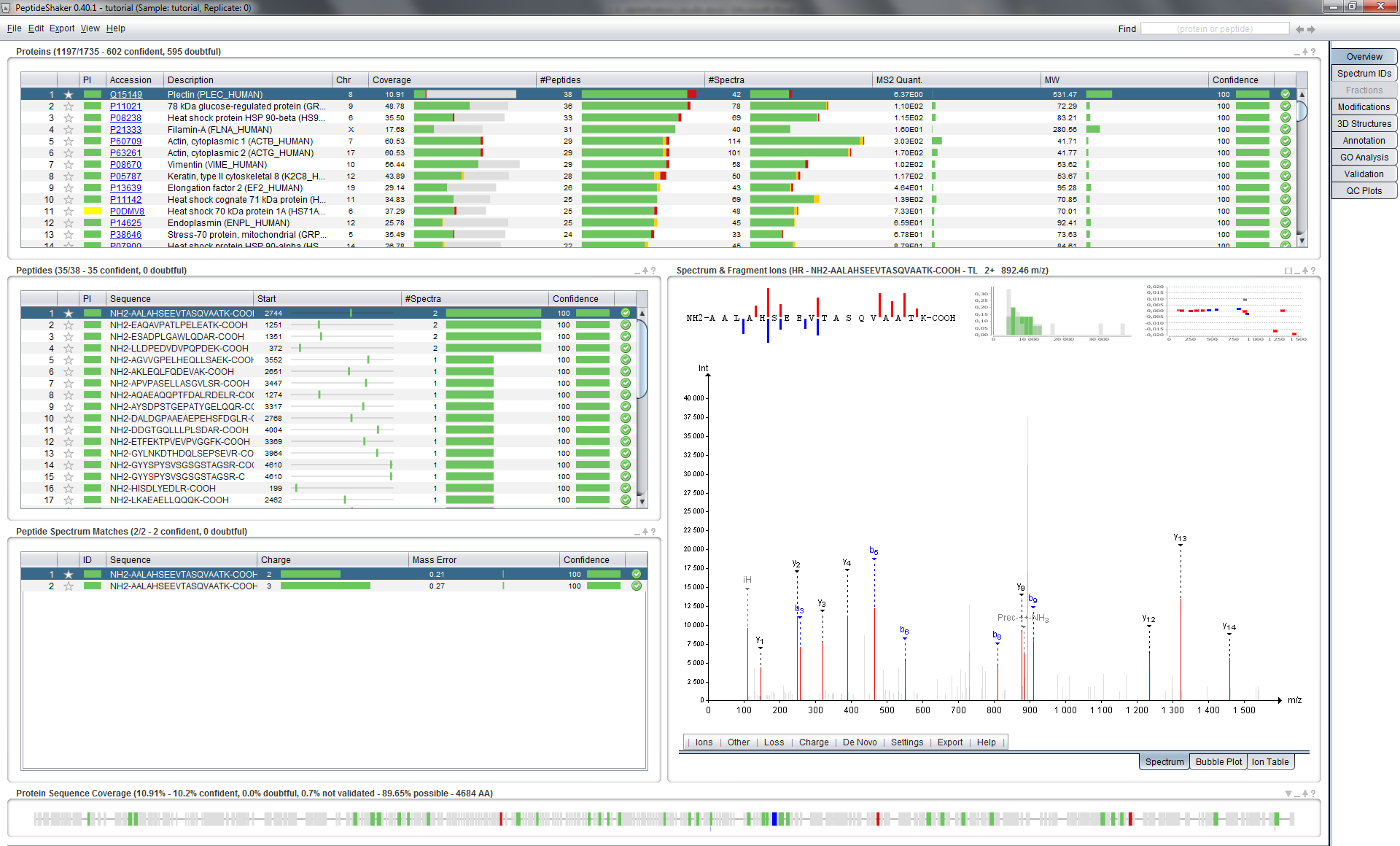
**Tips:**

*Do not put databases or mgf files on external drives, prefer a local SSD disc: the software needs to read these often, maximizing the reading speed will reduce the loading time!*

*Before starting an ambitious experiment, make sure that the software you need exists and that you have the computing power to analyse the data. Run smaller tests before starting the experiment!*

*Before loading a large dataset consisting of multiple runs, load a single run first to verify that everything works fine!*

When finished, the dialog closes and the identification results are displayed, you should see something like this:

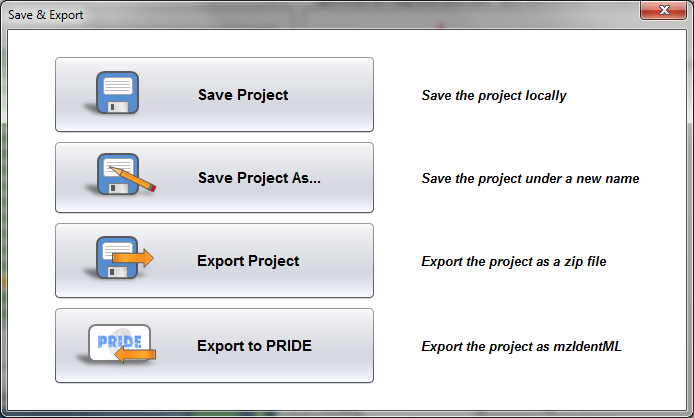


This is the main screen of PeptideShaker and it gives you an overview of the results in your files. PeptideShaker is split into various tabs found in the upper right corner, each with a different function.

The Overview tab is split into five main sections: the protein table at the top; the peptide and peptide to spectrum match (PSM) tables, both in the middle left; the spectrum view in the middle right; and the sequence coverage panel at the bottom. All five sections are connected. This means that selecting a protein in the protein table updates the peptide table, the PSM table, the spectrum view and the sequence coverage panel. Try selecting some of the other proteins in the protein list and see how the other sections are updated accordingly.

At the top right of every section there are small icons allowing you to interact with the data displayed: a square to maximize the section, an underscore to minimize it, an arrow to export the content and a question mark to open contextual help.

Before doing anything else we will start by saving our PeptideShaker project. This way we do not have to reload all the data the next time we want to look at them, but can rather do a much faster load of the project directly. To save your project, go to the ‘File’ menu and click the ‘Save As’ item. PeptideShaker offers you different save options:



‘Save’ and ‘Save Project As…’ allow you to save the project in the PeptideShaker format (.cps files). ‘Export Project’ exports the project together with all related spectrum and database files as a single zip file. This option is particularly interesting to share your results: send the zip file to a colleague, and after unzipping he or she will be able to open your project in PeptideShaker with just a few clicks and interact with your results. ‘Export to PRIDE’ allows you to save your project in the standard mzIdentML format, this will be the subject of the “Data Sharing” chapters. For now, click on ‘Save Project As…’, choose a name for your data file, and click ‘Save’. Note that saving can some take time: there is a lot of information to store!

A .cps file should now have been created. If you want to open this project later simply open this file via the PeptideShaker Welcome Dialog. The project can also be accessed via the 'Open Project' and ‘Open Recent Project’ options in the 'File' menu after starting PeptideShaker.

As already mentioned, protein identification using mass spectral data always starts with the spectra. We will therefore start by looking at a particular spectrum in detail and see how well this corresponds to the identified peptide.

Start by making sure that the first protein in the protein table is selected, the one with the protein accession number Q15149, descriptionPlectin (PLEC\_HUMAN) and originating from Chromosome 8.

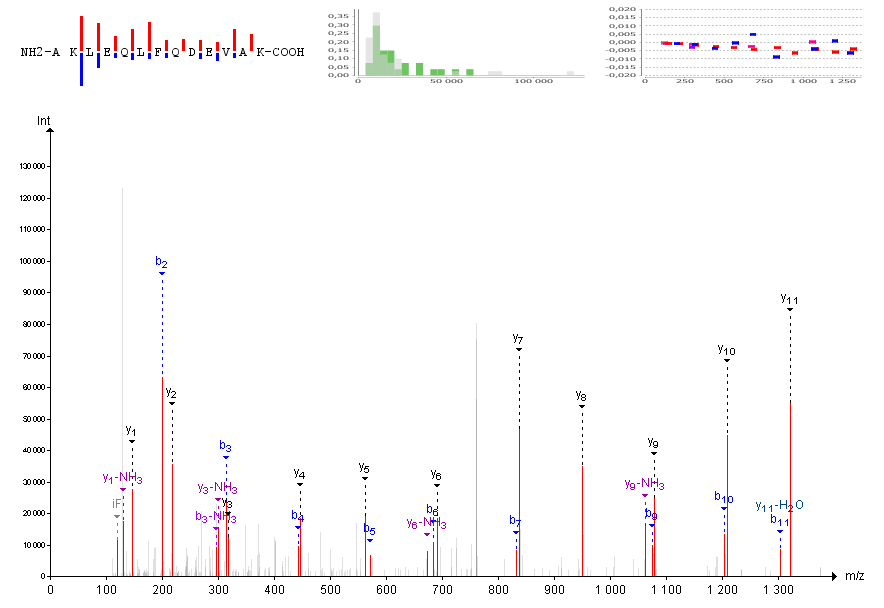
Looking at the row for this protein we see that we have covered 10.91 % of the protein's amino acid sequence, detecting a total of 38 peptides from 42 spectra. (Note: The color coding for the peptides and spectra will be explained in the “Peptide and Protein Validation“ chapter.) *Why are there more spectra than peptides*? *How do you define a peptide?* *[1.4c]*

In the peptide table, you will see that some amino acids are colored. These residues were identified as carrying a post-translational modification (PTM). In fact, for an easy interpretation, PTMs are color coded everywhere in PeptideShaker. Holding the mouse over a colored residue will give you the PTM details. *How many modified peptides were identified for this protein?* *[1.4d]*

**Tip:**  
*By default only variable modifications are displayed. You can change this in the ‘View’ menu via the ‘Fixed Modifications‘ option.*

Make sure that the first peptide sequence AALAHSEEVTASQVAATK is selected. In the peptide to spectrum matches table you will see that this peptide is found two times. That means that two different spectra have been identified as the peptide in question. Also notice that one has a precursor charge of 2+, while other has a charge of 3+. *Why do not all precursors carry the same charge?* *[1.4e]*

Now move to the peptide sequence AKLEQLFQDEVAK. The spectrum used to identify the peptide is now shown to the right of the table, and should look like this:



*By looking at the spectrum, is it obvious that this spectrum is identified as the sequence* AKLEQLFQDEVAK*?* *Which are the different ions displayed here? Are the same ions detected in all peptide to spectrum matches or peptides? What are the standard fragment ion types and how do they relate to the peptide sequence?* *[1.4f]*

The menu bar below the spectrum allows you to fine tune the annotation settings and the spectrum display. It is additionally possible to export the spectrum in various formats. Note that only the highest peaks are annotated. You can change the annotation level by scrolling when over the spectrum. If you hold ‘Ctrl’ and scroll, the m/z tolerance will be changed.

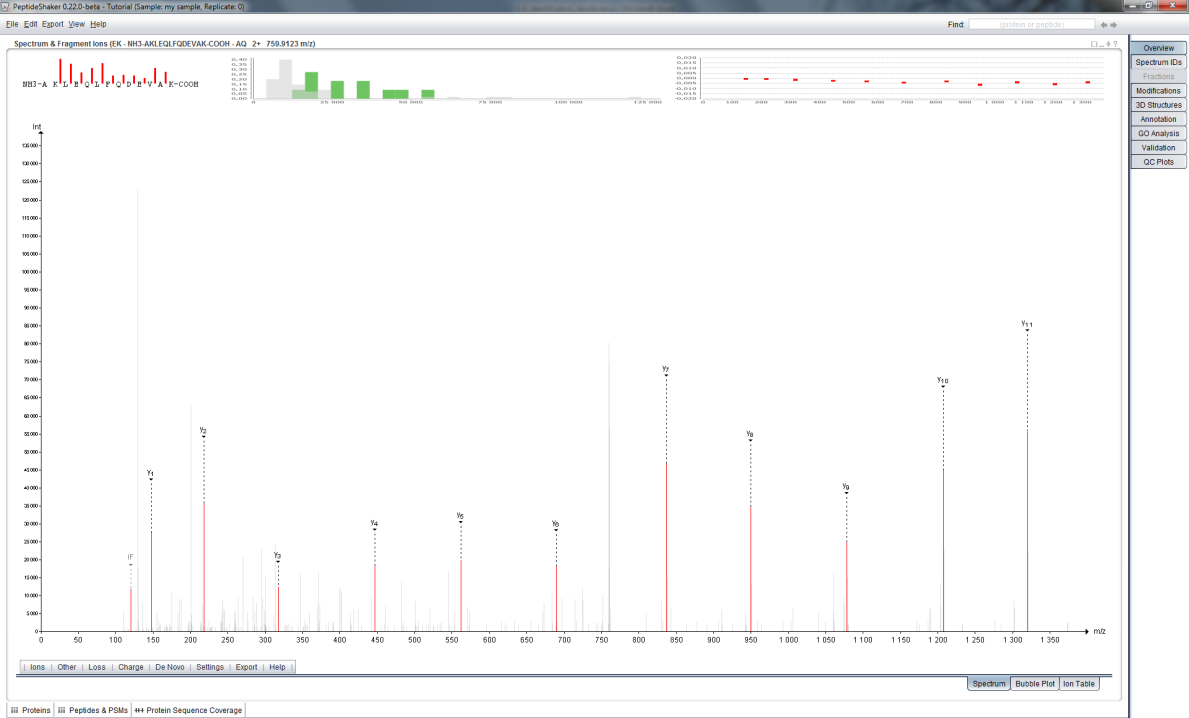
**Tip:**  
*You can also change the annotation tolerances using sliders which can be enabled in the ‘View’ menu.*

Before continuing, make sure that the intensity level is at 75% and the accuracy at 0.02 Da. (The values can also be set via *Edit > Spectrum Annotations*.)

The first thing we will do is to see how well the spectrum matches the sequence it has supposedly identified. To do this we will perform a bit of manual *de novo* sequencing, essentially checking whether the detected fragment ions match the peptide sequence. We will start with the y-ions, so disable the display of b-ions by clicking the 'Ions' menu below the spectrum and unselecting the ‘b’ ion in the list.

Neutral losses are selected automatically by PeptideShaker depending on the peptide sequence and modification status of the peptide. In order to hide the fragment ions with a neutral loss, unselect ‘Adapt’ in the ‘Loss’ menu, and then deselect the neutral losses one by one.

To make the sequencing a bit easier we will also hide the other sections of the Overview tab. Click on the square in the top right corner of the spectrum panel. Note that the other tabs have been minimized at the bottom left of the screen. The spectrum should now cover the whole screen:



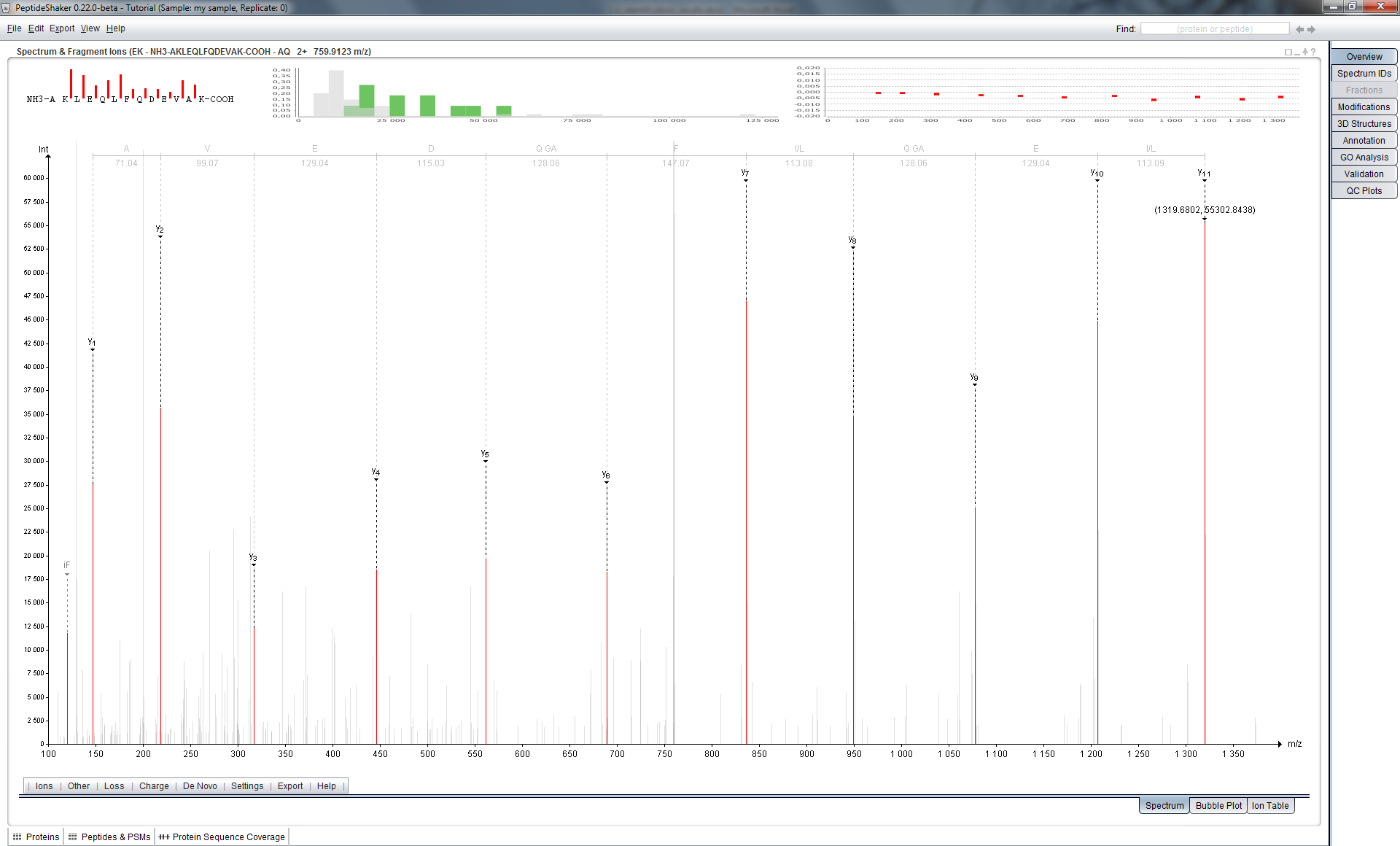
Next, to make the sequencing even easier, zoom in on the m/z range covered by the y-ions y1 to y11, simply by clicking and holding the left mouse button from just before the y1-ion and (while holding the left mouse button pressed) dragging the mouse horizontally to the right until you have passed the y11-ion. At that point just let go of the left mouse button to zoom in on the selected range. If you want to zoom in further, you can repeat the same procedure. If you at any point want to zoom out again, simply click the right mouse button to return to the full spectrum display.

Now start the *de novo* sequencing by first clicking the y1-ion. If you still find it difficult to click the correct peak, try zooming in just on the peak in question, select it, and then zoom back out again.

**Tip:**  
*In order to click a peak, it first needs to be highlighted (showing its exact m/z and intensity above the peak) and this highlighting only occurs if the mouse is held above the peak within 1.5 times the peak height.*

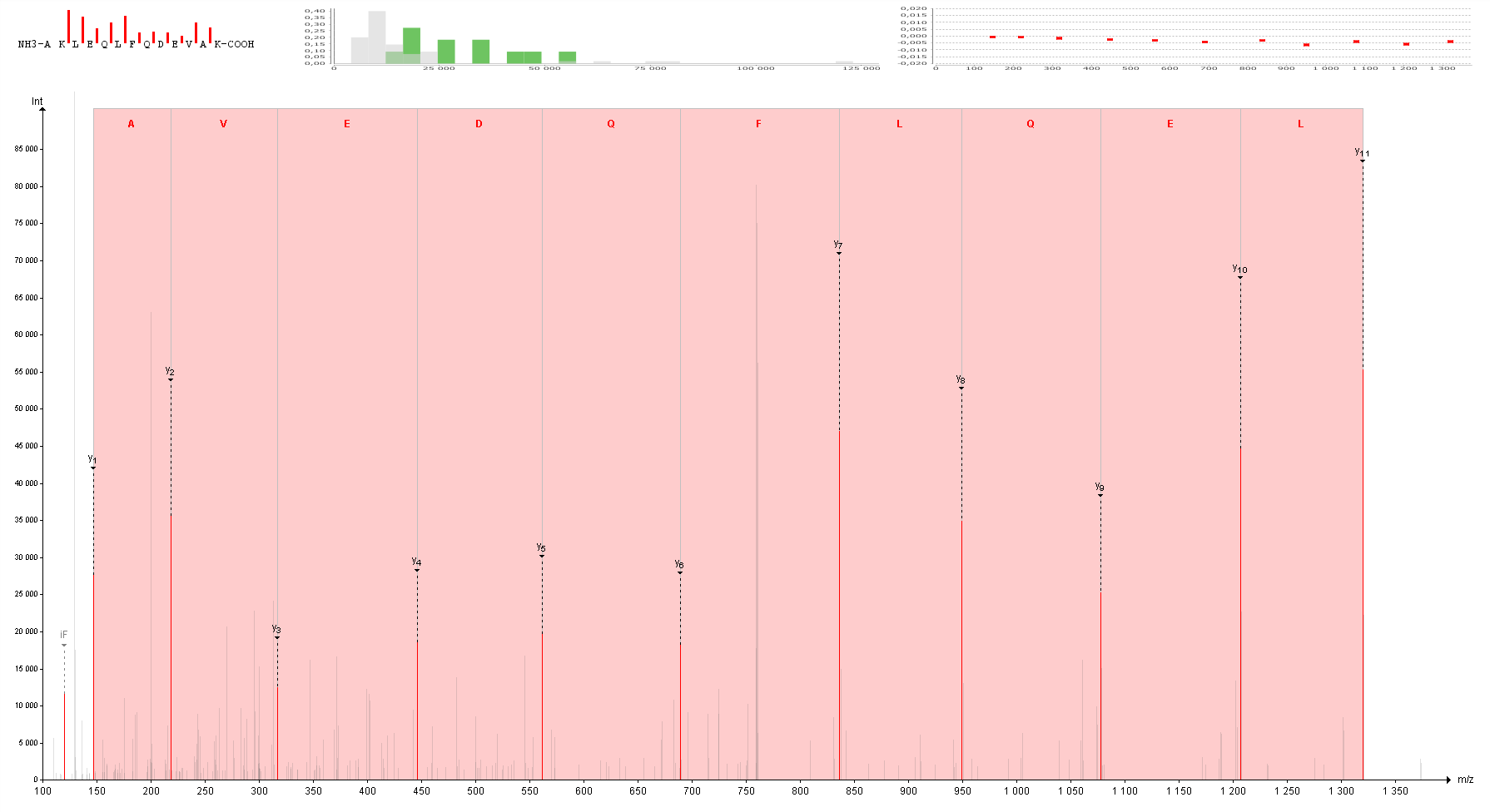
Continue the sequencing by clicking the y-ions in ascending order from left to right. If you click the wrong peak, the last selection can be removed by holding down the Shift key while clicking the peak again. Ctrl + Click will save the current sequence, and will allow you to start a second sequence.

When done, you should have a spectrum display that looks something like this:

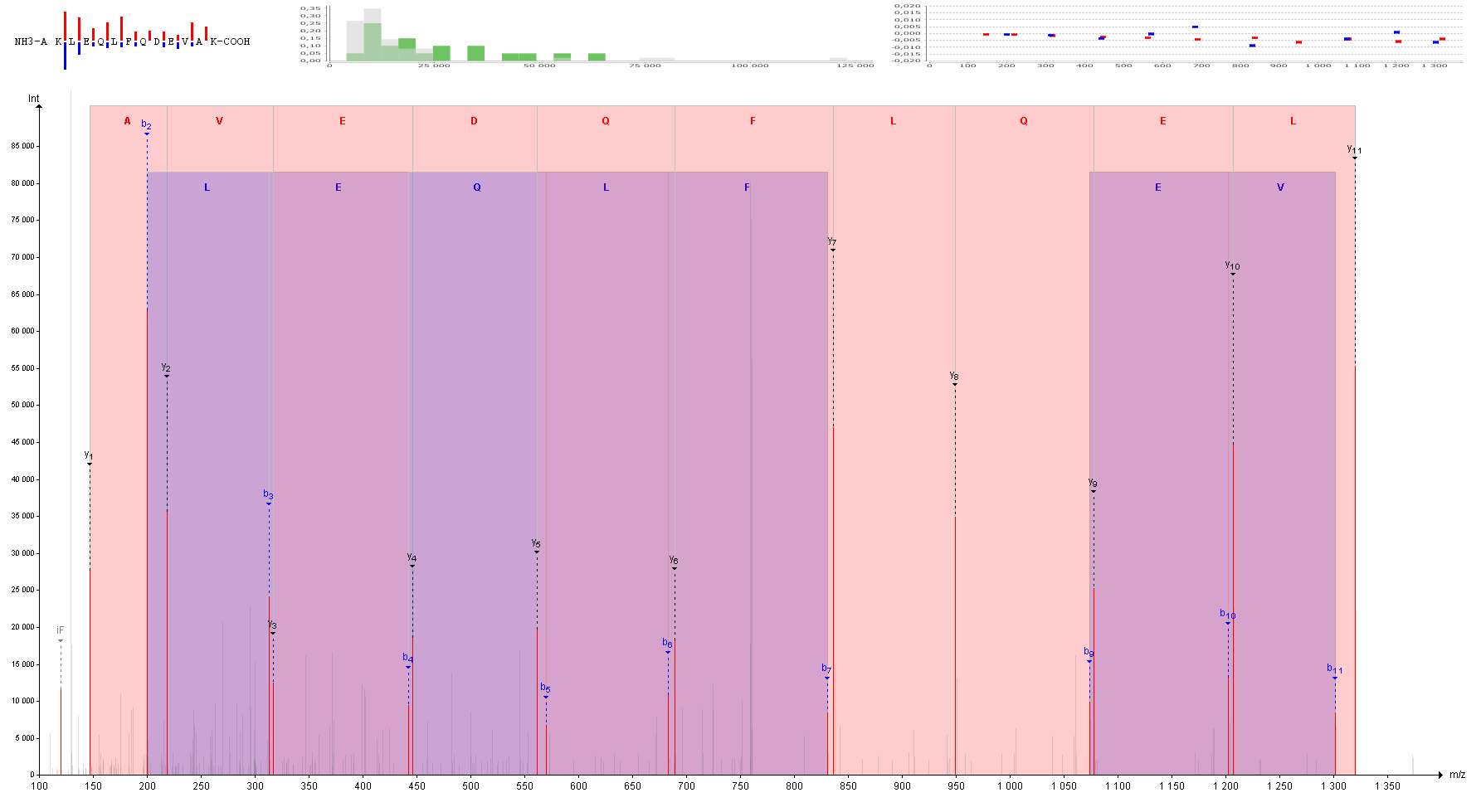


If you now read the sequence at the top you will find that it reads (from left to right): A, V, E D, Q GA, F, I/L, Q GA, E, I/L. Given that y-ions are indexed from right to left (carboxy-terminus to amino-terminus) relative to the sequence, we have to reverse the sequence to match it to the peptide sequence. Hence we get the sequence: I/L E Q/GA I/L F Q/GA E V A. Comparing this to the peptide sequence we see that they are very similar, and if we resolve the ambiguous residues, we have a perfect match to the proposed peptide's sequence from residue 3 - 12. *Why do we not have complete coverage? And why is complete coverage in most cases not necessary? Where do the ambiguous residues come from? Can they impact the final result? What is the role of modifications in the ambiguity? Can you relate this to identification issues when many (variable) modifications need to be considered? What is an immonium ion, and how can these be used in the de novo sequencing?*  *[1.4g]*

In the ‘De Novo’ menu, select ‘y-ions’, PeptideShaker then shows you the solution retained for this spectrum:



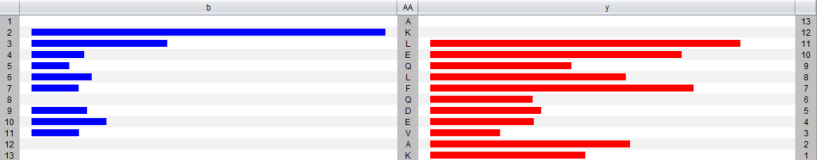
Now, enable the display of b-ions and select ‘b-ions’ in ‘De Novo’. You should see the following:



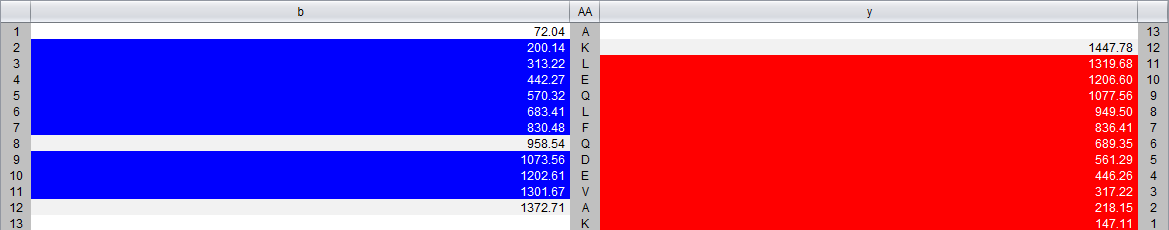
Note that for the b-ions you can only use b2 to b7 and b9 to b11. Remember that b-ions are indexed from the left to the right (amino-terminus to carboxy-terminus), so in this case you do not have to reverse the sequence. *Why do we have lower coverage and intensity for b-ions relative to the y-ions for the same peptide? Is this the same for all peptides, all instruments, all protocols?* *[1.4h]*

When you are done sequencing, make sure that both the b- and the y-ion types are selected in the menu below the spectrum, and click the ‘Ion Table’ tab in the lower right corner.

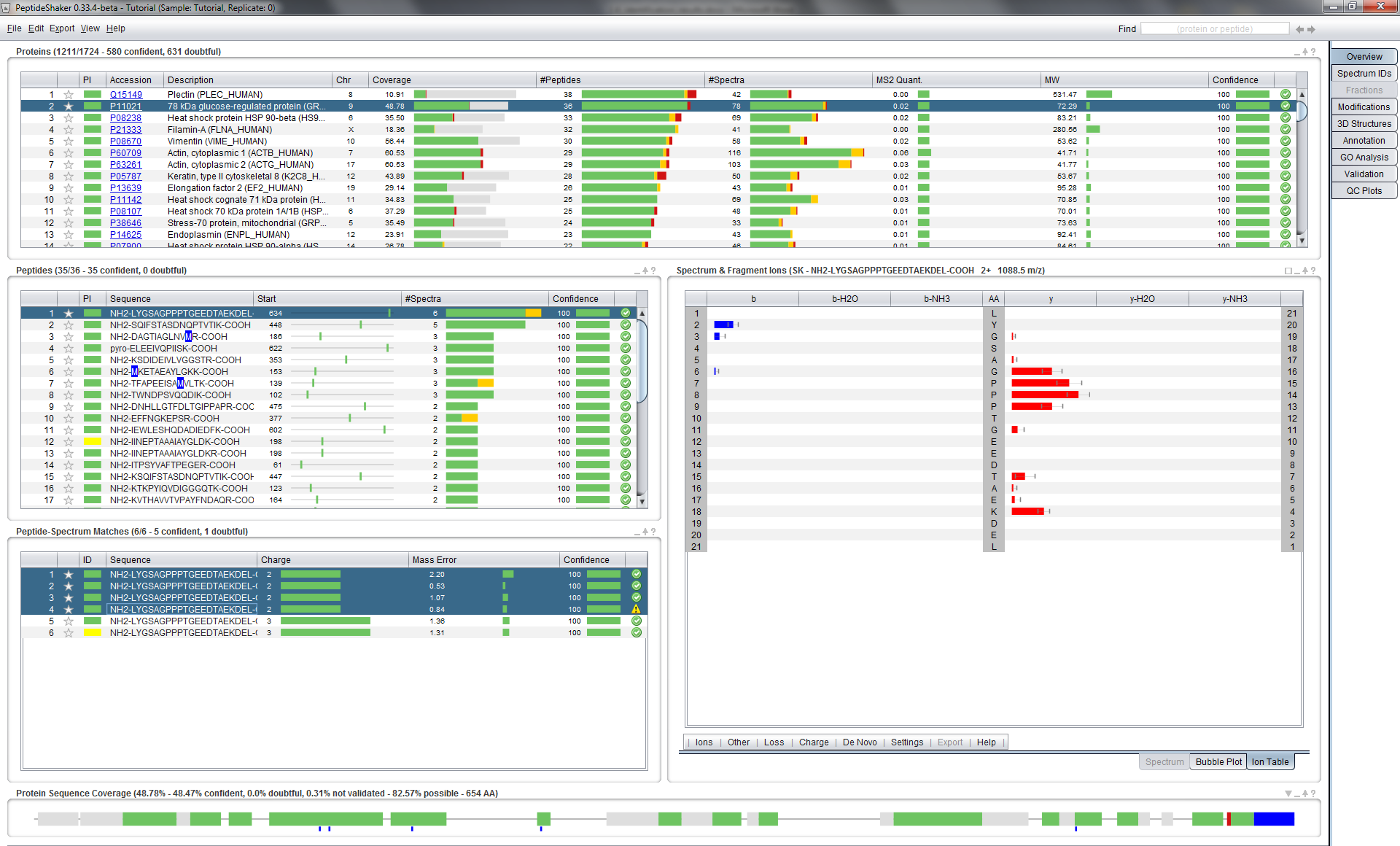
Here you will see an overview of the detected fragment ions and how they correspond to the sequence of the peptide. *How do the results here correspond to the results you found in your de novo sequencing?* *[1.4i]*



In the 'Settings' menu below the table you can also select m/z based display instead of intensity:



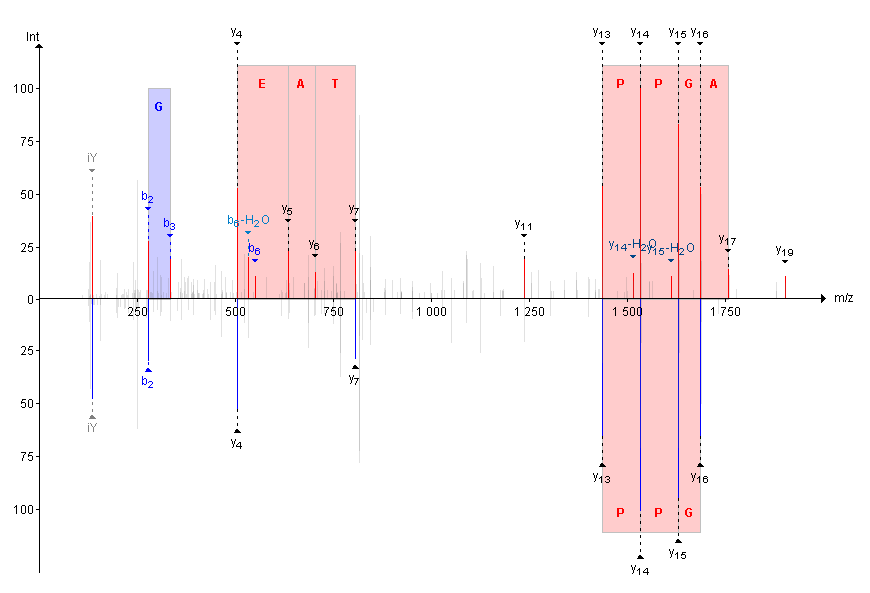
We will now move one step further by comparing multiple spectra identified as the same peptide. Bring the spectrum panel back to its normal size by clicking on the square and select the second protein in the protein list: P11021. You will see that the first peptide (LYGSAGPPPTGEEDTAEKDEL) has been identified by six spectra: four with a charge of 2+ and two with a charge of 3+. While holding down the Ctrl-key, select all spectra with a charge of 2+. You should see this:



The intensity difference between the spectra is displayed with error bars. *Do you expect reproducibility in intensities between spectra? What do you think about the comparison between 2+ and 3+ PSMs?* *[1.4j]*

In a standard visual inspection of both spectra (select them one by one in the PSM table and look at each spectrum) it can be difficult to tell how similar they are. Attempting to see which fragment ions are detected in all of the spectra, and at which intensities, is also very hard.

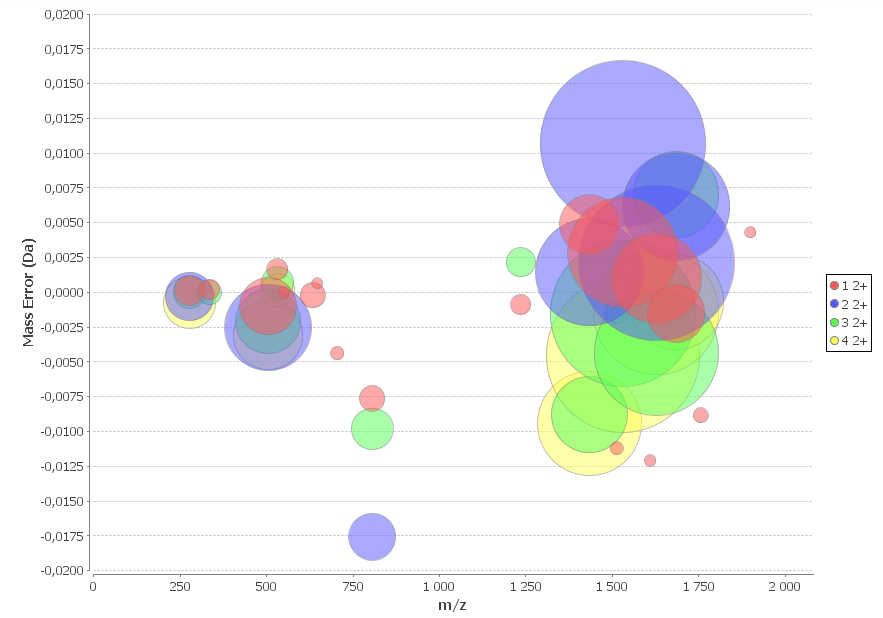
If comparing two spectra so-called mirrored spectra can be used, i.e., showing one spectrum above the x-axis and another spectrum mirrored below the x-axis. Select only the two first PSMs in the table and go back to the Spectrum tab. You should now see the following:



The use of mirrored spectra allows you to easily compare two spectra and highlight the differences between them. (Note that in order to make the intensities comparable the peaks are shown relative to the highest peak in each spectrum.)

When comparing more than two spectra PeptideShaker uses a novel view, referred to as the bubble view or the *Planetary System View*. You will find this view by clicking the ‘Bubble Plot’ tab below the spectrum (or by simply selecting more than two rows in the spectrum table). Selecting/highlighting all the rows in the PSM table with a 2+ charge should give you something like the figure below.

In this view of the spectra we still have the m/z value on the x-axis, but on the y-axis we now have the mass error (the distance between the theoretical and the experimental mass of the fragment ions). The size of each bubble represents the (normalized) intensity of each fragment ion.



This way of looking at multiple spectra takes a bit of getting used to, but once one is comfortable with the setup, it becomes very easy to pick up details that would have been very difficult to see by a manual inspection of each spectrum individually. *Now, what do you think about the error distributions?* *[1.4k]*

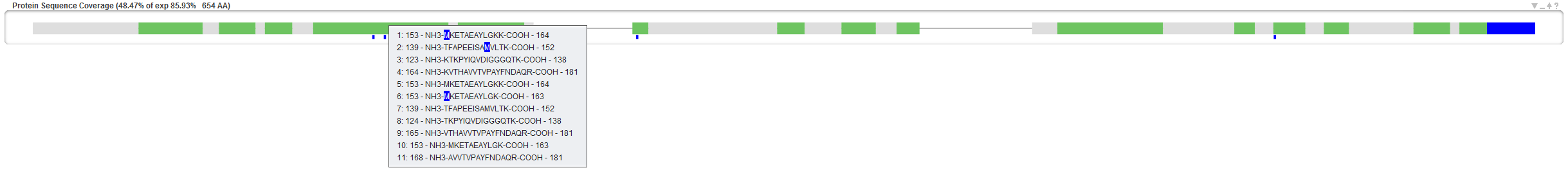
We will now study the sequence coverage of the selected protein – P11021. The sequence coverage is indicated in the protein table, 48.78% and illustrated at the bottom in the Protein Sequence Coverage panel:



The 36 identified peptides are mapped onto the sequence in green, yellow and red (color coding will be explained later), whereas the non-covered parts are left in grey. Note that the parts of the sequence which do not generate observable peptides are thinner than the others. For this protein, the observable coverage was estimated to 82.57% - the observable coverage is also displayed in gray in the protein table.

If you hold the mouse over a part of the sequence, the observed peptides will be displayed. *The currently selected peptide is displayed in blue, what is the first peptide observed in the sequence?* *[1.4l]*

Note that this peptide contains a missed cleavage and that the cleaved version is also detected. Also, keen observers will have noticed that the modifications are readily mapped under the sequence. Clicking at the corresponding peptide section allows the selection of the oxidized peptide:



*How many modified peptides does this protein contain? How many modification sites?* *[1.4m]*

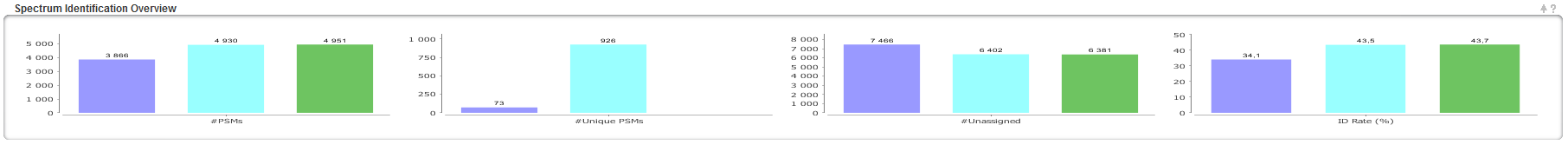
More details on PTM localization will be given in the upcoming “PTM Analysis” chapter.

1. Quality Control

Before further investigation, it is important to verify the quality of the search. We will start by verifying that all search engines worked correctly. As you remember, we loaded identification results from X!Tandem and OMSSA. Open the tab called ‘Spectrum IDs’ in the upper right corner, loading a tab can take a few seconds.

**Tip:**  
*When PeptideShaker is busy with a process its icon turns orange. When finished it turns grey again to inform you that the coffee break is over!*

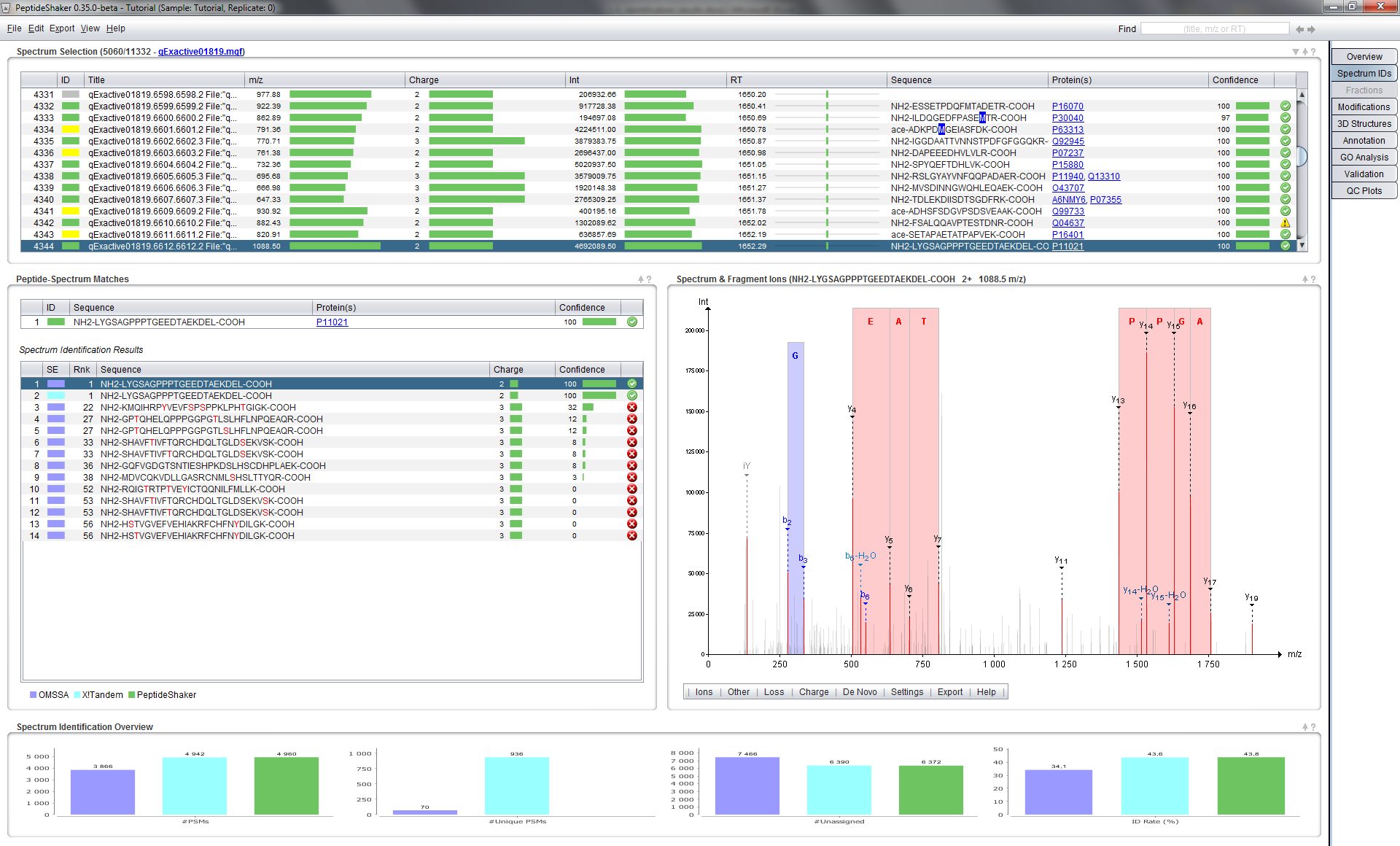
At the bottom of the tab you will see something like this:



Here you see how the search engine results compare at the PSM level. Bar charts that for each search engine shows the number of validated PSMs, unique PSMs, unassigned spectra and the ID rate.

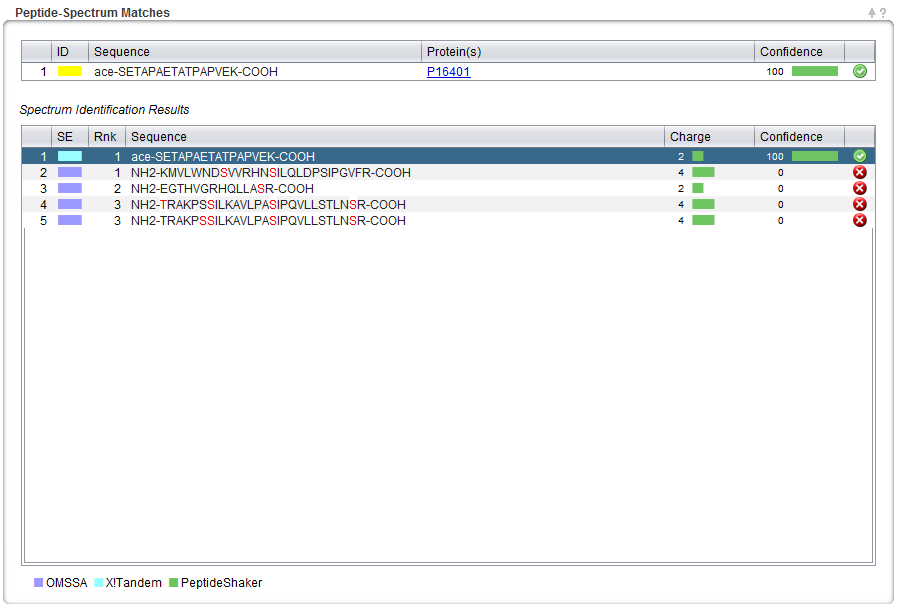
*Why are the search engine results different? Will the differences depend on the dataset used? Can we take advantage of these differences?* *[1.4n]*

The rest of the ‘Spectrum IDs’ tab allows you to browse the search engine results and see how PeptideShaker combines the separate search engine results into a single result set. By clicking a spectrum in the spectrum table at top you will see the PSMs for the search engines used and how these were scored. The selected PSM will be used to annotate the given spectrum.



**Tip:**  
*If you are interested in non-assigned spectra, you can export them using the ‘Follow Up Analysis’ option in the ‘Export’ menu.*

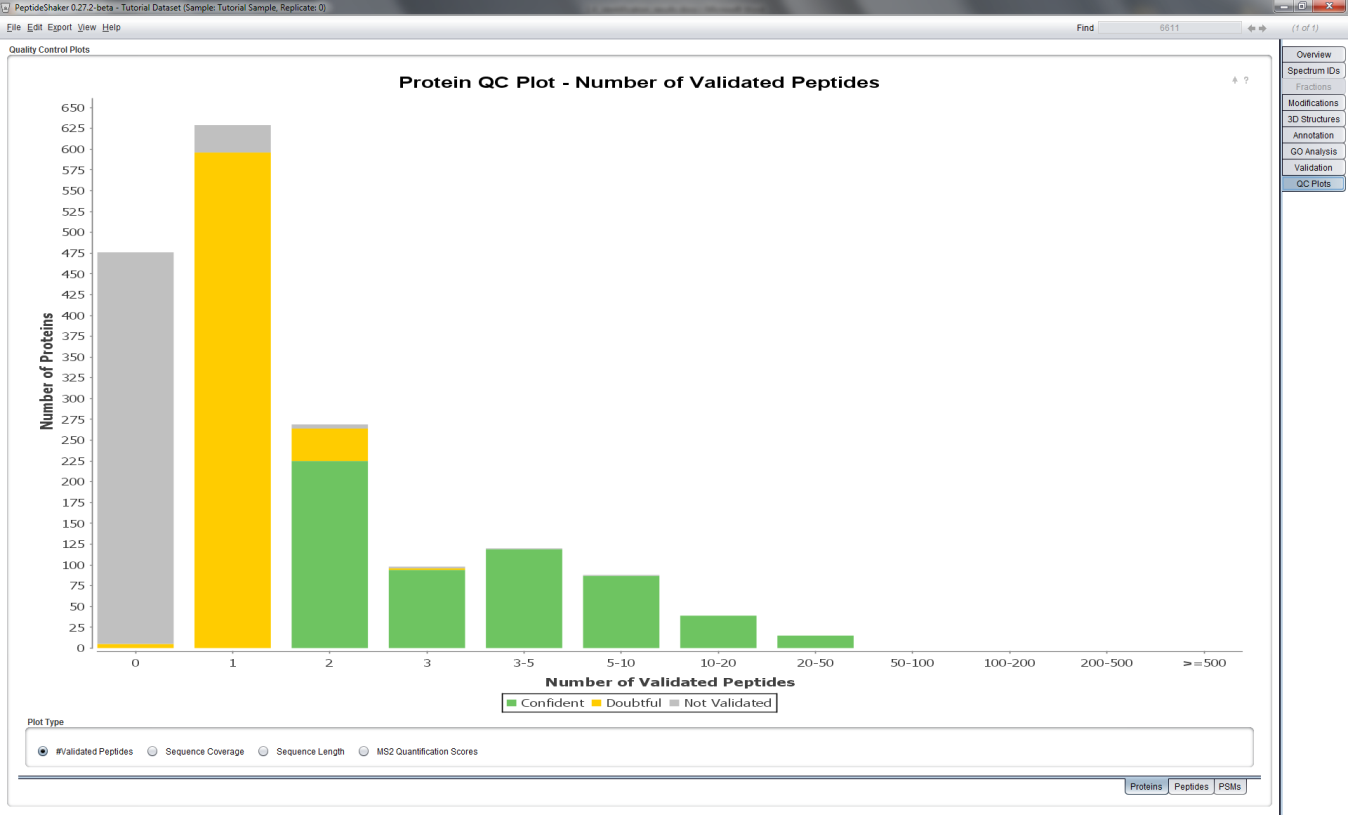
Note the ID column which indicates whether a spectrum was identified and if so, whether the search engines agreed on the identification. Select row 4343, spectrum qExactive01819.6611.6611.2 File:"qExactive01819.raw", NativeID:"controllerType=0 controllerNumber=1 scan=6611", with a yellow square. In the Peptide-Spectrum Matches section you should see this:



As you can see, a peptide found by X!Tandem took precedence over others suggested by OMSSA. *Which of the possibilities is the best match in your opinion?* *Would you trust peptides where the search engines disagree?* *[1.4o]*

Now, select the ‘QC Plots’ tab. In this tab, a list of Quality Control metrics are given for the proteins, peptides and PSMs. The category (Proteins, Peptides, PSMs) is selected at the bottom right corner of the screen.

The default view is the Protein option, and you should now see the distribution of proteins according to their amount of identified peptides:



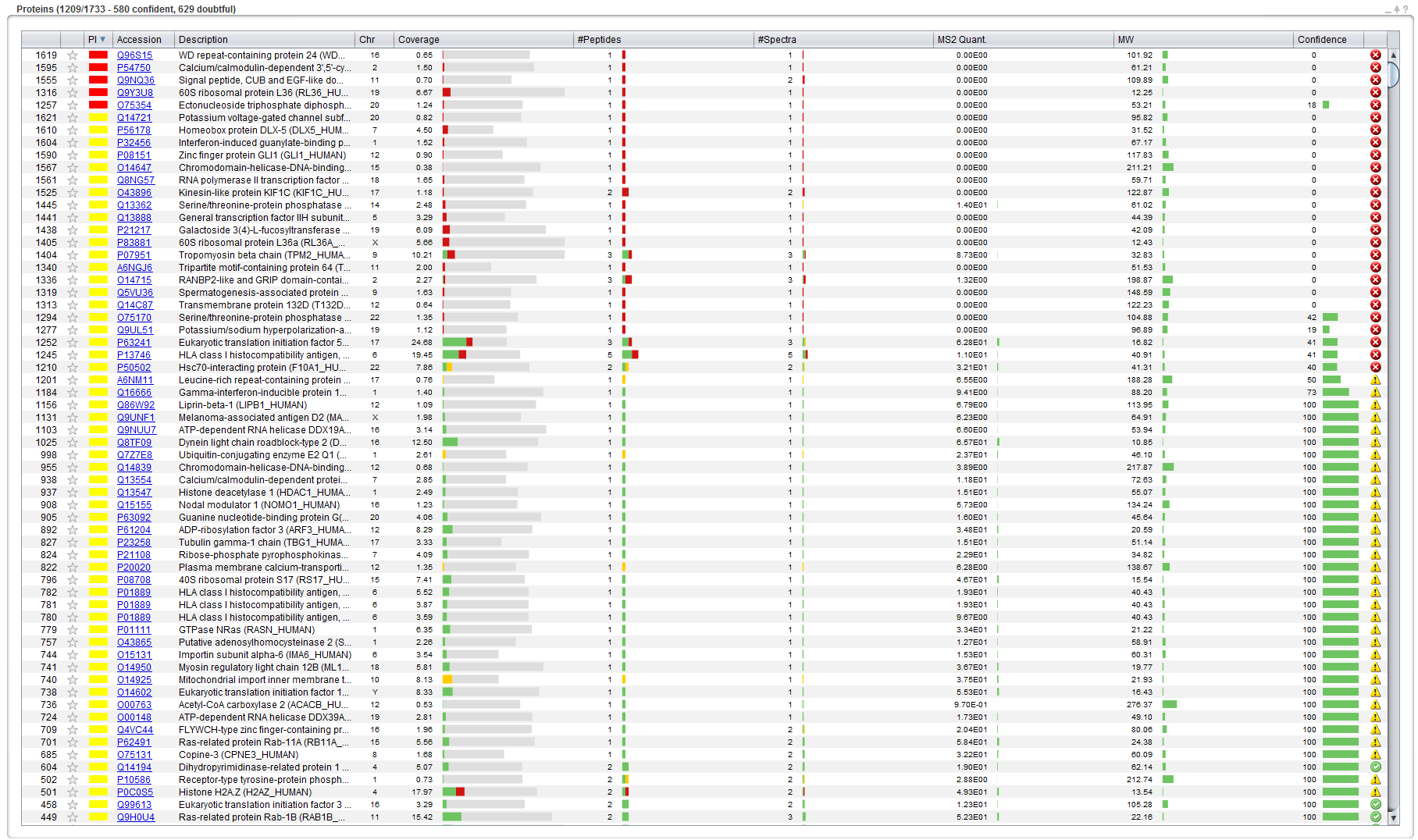
*How many peptides would you require to trust a protein identification?* *[1.4p]*

To save time, we will not go through all quality control plots but feel free to explore them by yourself. We welcome any questions/suggestions!

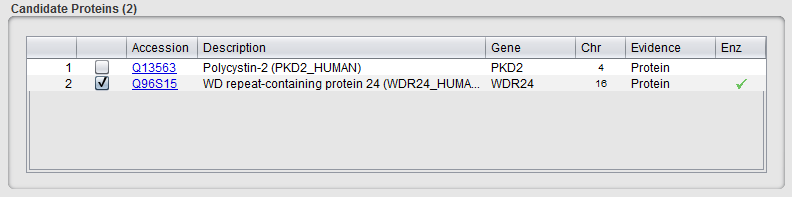
1. Protein Inference

When building proteins from peptides, it is in some cases impossible to decide between two proteins that share the same peptide(s). This issue is known as the *protein inference problem*.[3](#_ENREF_6) Go back to the ‘Overview' tab and sort the proteins by the PI column by clicking the column header, and click the header once more to sort in descending order. PI stands for Protein Inference, and the proteins are now sorted so that the proteins with protein inference issues are at the top.

In PeptideShaker, proteins are flagged using four colors: green, yellow, orange and red. For the best overview, hide all sections but the Protein table:



Click on the red box corresponding to protein accession number Q96S15, and a Protein Inference Dialog appears, displaying two proteins (we will come back to the protein inference graphs later). PeptideShaker could not decide whether Q96S15 or Q13563 was present in the sample as the identified peptide is shared by the two proteins. *According to the information provided, which protein suggestion would you trust? What do the Evidence and Enz columns represent, and how can this information be used? [1.4q]*



Note that proteins in a protein group can be attached to different genes or chromosomes. Choosing one or the other can hence strongly impact the biological interpretation of the results! PeptideShaker always chooses the protein with the best evidence according to UnitProt.[4](#_ENREF_7)

**Tip:**  
*In order to benefit from comprehensive protein annotation, we strongly recommend the use of UniProt databases.*

Inferring proteins from peptides is not as simple as it may seem. Some peptides can map to different proteins, let’s say protein A and protein B. If no unique peptide unambiguously indicates the presence of either protein A or B, PeptideShaker cannot say whether protein A or protein B is present.

Confidence

A

B

AB

B

AB

A

A

B

AB

Case 3

Threshold

Case 1

Case 2

*In this simple example, how do you interpret the various cases? [1.4r]*

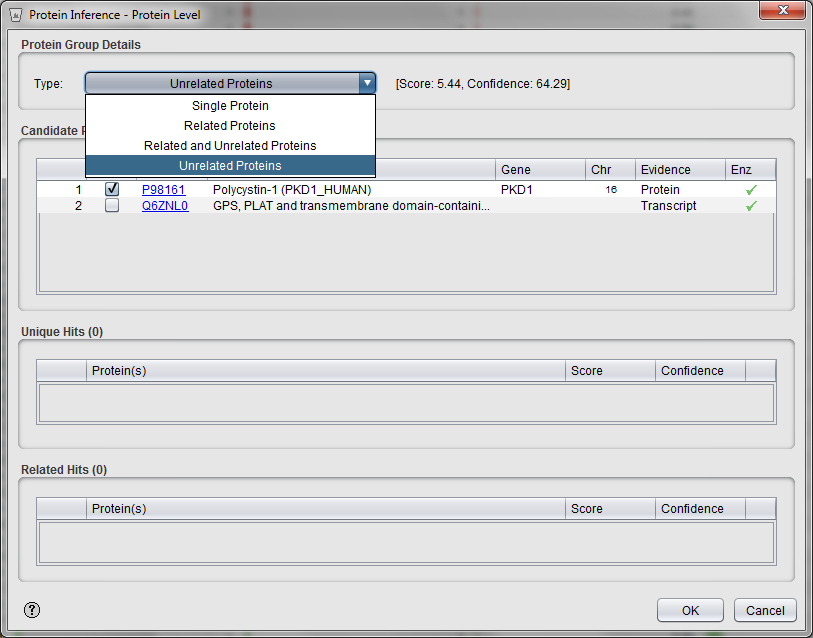
Often, the problem is a lot more complex and involves dozens of intricate peptide to protein mappings. PeptideShaker solves protein inferences of Case 1 and 2, as you may have noticed when loading the identification files. However, Case 3 requires additional information about the sample to find out, if possible, which protein was actually present. These types of conflicts are in the PI column colored in yellow, orange and red depending on the properties of the conflict. By clicking on the corresponding box (as we have done above), you can manually select the protein you consider as the correct one.

In most cases, the protein groups consist of related proteins. Based on the protein gene name and the protein descriptions, PeptideShaker sorts the protein inference conflicts into four categories:

* Green: No Conflict, i.e., a Single Protein unambiguously identified
* Yellow: Group of related proteins
* Orange: Group of Related and Unrelated Proteins
* Red: Group of Unrelated Proteins

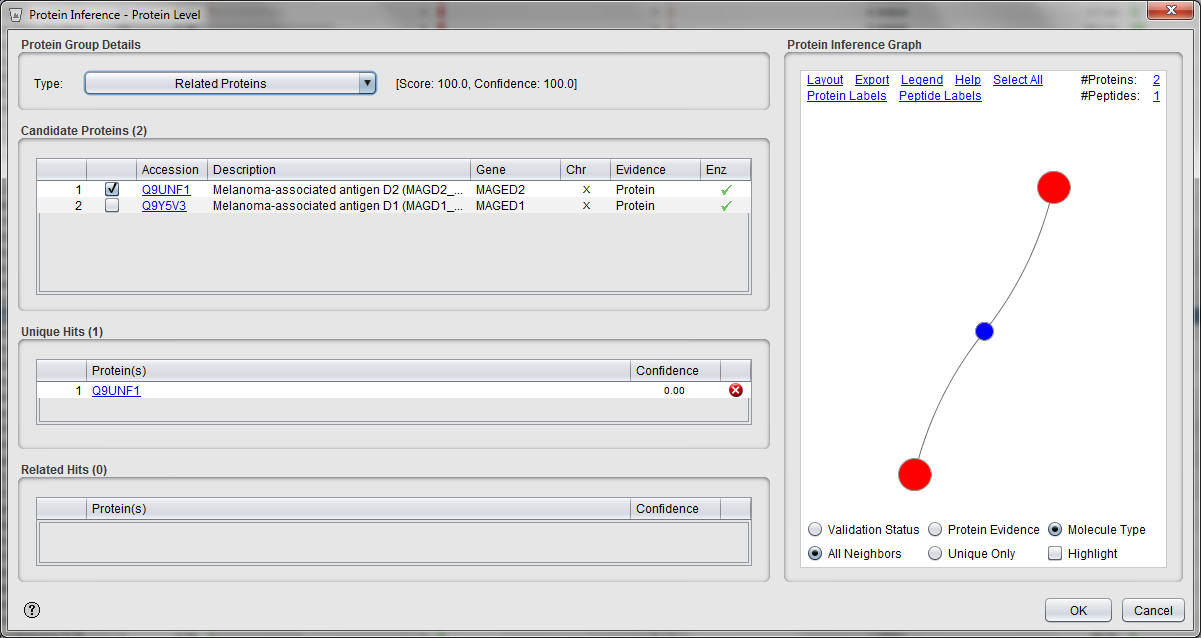
*How accurate is this type of sorting?* *[1.4s]*

The conflict type can be edited in the Protein Inference Dialog:



If the type is changed, the color for the group is updated in the PI column of the protein table.

Now look for protein Q9UNF1, and click on the yellow box in the PI column. The dialog below will appear:



Here, a peptide belonging to Q9UNF1 was found shared with Q9Y5V3, and the group was detected with a high confidence (100%). Additionally, the dialog also shows that Q9UNF1 was also identified by unique peptides but with a very low confidence. As a result, Q9UNF1 will appear twice in the protein table: once as a group (Q9UNF1 or Q9Y5V3) and once as unique hit (Q9UNF1).

In order to navigate such hits, we will use the Find feature of PeptideShaker: type Q9UNF1 in the top right corner of the interface.



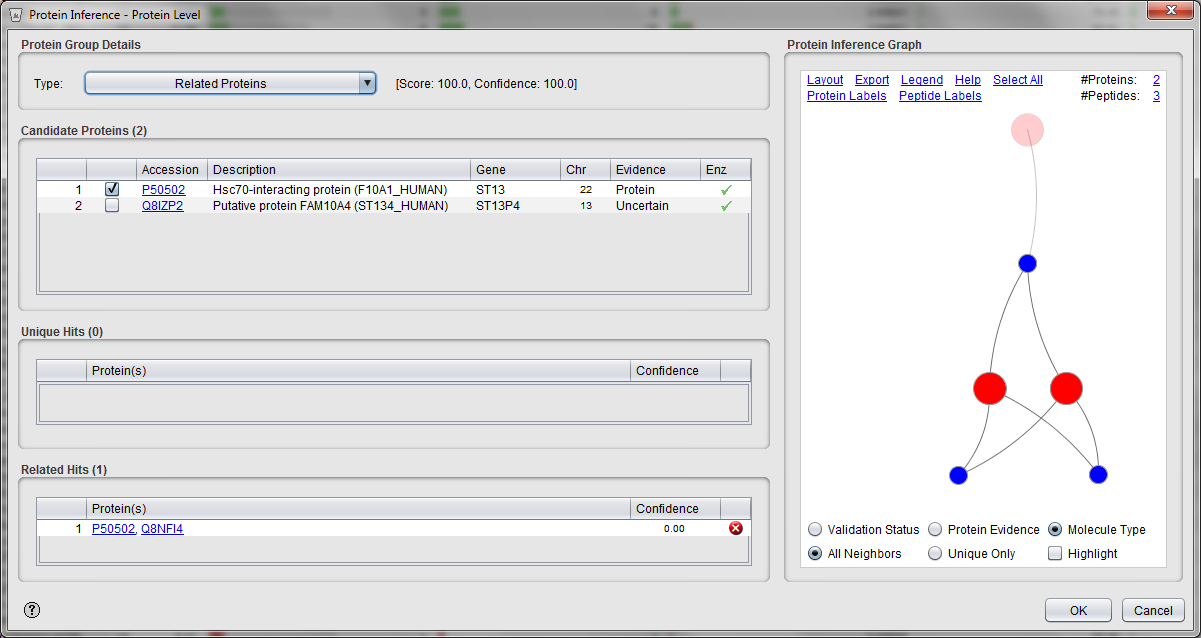
Using the two arrows, you can navigate the two groups related to this accession. *Select the unique protein hit, can you find the unique peptide?* *[1.4t]*

Note that you can change the retained accession for any group by selecting the protein in the first column in the Protein Inference dialog.

*Can we delete a useless group?* *[1.4u]*

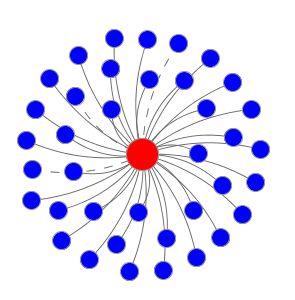
**Tip:**  
*The Find function accepts protein accessions, descriptions and even peptide sequences!*

Similarly, if peptides shared with one of these proteins were found in another group, the group is reported in the 'Related Hits': for the group (P50502 or Q8IZP2), P50502 is shared with the group (P50502 or Q8NFI4):



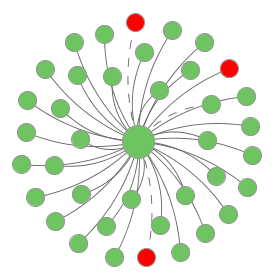
As you have already seen examples of above protein inference can be displayed as graphs. We will now look at some examples and see how such graphs can help better understand the complexity of protein inference.

First, sort the proteins on the index column (the first column in the protein table) in ascending order by clicking the column header. Now select the first protein Q15159 - Plectin (PLEC\_HUMAN) and click the green square in the PI column. This will show the Protein Inference Dialog where you will see the following protein inference graph:

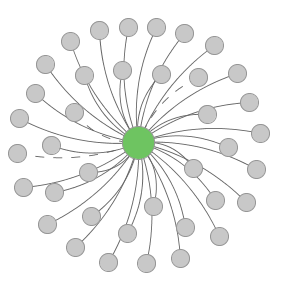


The big red circle represents the protein while the blue smaller circles represent the peptides. A line is drawn between a peptide and a protein if the sequence of the peptide can be located in the sequence of the protein. A fully-drawn line means that the peptide sequence is enzymatic in the given protein, while a dotted line means that the peptide sequence is not adhering to all the cleavage rules of the enzyme used. *How many non-enzymatic peptides have been detected for the protein above?* *[1.4v]*

Now we will add some additional annotation on top of the graph. Select the Validation Status option below the graph. This will color the proteins and peptides depending on their validation status (from Confident (green), *via* Doubtful (yellow) to Not Validated (red). (The validation details will be further explained in the next chapter.) *How many doubtful and non-validated peptides have been detected for the given protein?* *[1.4w]*

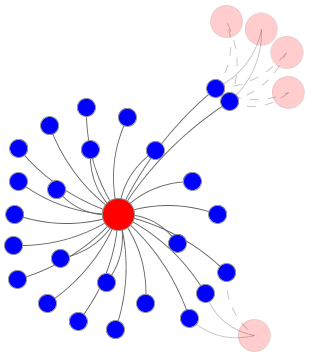


If a UniProt database is used it is also possible to color code the proteins based on the protein evidence available in UniProt (from Protein Evidence (green) to Uncertain (red)):



For an overview of the color coding options, click the Legend option at the top of the graph.

Now we will look at a slightly more complex example. Open the Protein Inference Dialog for the protein PP14625 - Endoplasmin (ENPL\_HUMAN).



The select protein in the middle represents Endoplasmin (as you can confirm by holding the mouse over it), while there are also five other proteins that could be used to explain some of the peptides attributed to Endoplasmin. *By using the graph options, can you figure out why the Endoplasmin protein was retained? Do you agree with this decision?* *[1.4x]*

If you have time, explore the protein inference graphs for other protein groups. *Can we avoid protein inference issues in shotgun proteomics?* *[1.4y]*

References

1. Vaudel, M. et al. PeptideShaker enables reanalysis of MS-derived proteomics data sets. *Nat Biotech* **33**, 22-24 (2015).

2. Martens, L. et al. PRIDE: the proteomics identifications database. *Proteomics* **5**, 3537-3545 (2005).

3. Nesvizhskii, A.I. & Aebersold, R. Interpretation of shotgun proteomic data: the protein inference problem. *Mol Cell Proteomics* **4**, 1419-1440 (2005).

4. Magrane, M. & Consortium, U. UniProt Knowledgebase: a hub of integrated protein data. *Database : the journal of biological databases and curation* **2011**, bar009 (2011).