Structure, Volume *22*

Supplemental Information

Mass Spec Studio for Integrative Structural Biology

Martial Rey, Vladimir Sarpe, Kyle M. Burns, Joshua Buse, Charles A.H. Baker, Marc van Dijk, Linda Wordeman, Alexandre M.J.J. Bonvin, and David C. Schriemer

Mass Spec Studio for Integrative Structural Biology

Martial Rey, Vladimir Sarpe, Kyle Burns, Joshua Buse, Charles A. H. Baker, Marc van Dijk, Linda Wordeman, Alexandre M.J.J. Bonvin, David C. Schriemer

Address correspondence to: dschriem@ucalgary.ca

List of supplementary elements

Architecture

Mass Spec Studio is written in C# using the .NET 4.0 framework. To allow for easy extensibility, the Studio employs Microsoft Extensibility Framework (MEF) and Prism (**Figure S1**). MEF is used to support easy third-party plugin extensibility and an underlying degree of modularity. Prism offers dynamic composition of user interface modules. Mass Spec Studio extensibility points for building packages can be divided into 4 categories: Experiment Types, UI Modules, Algorithms, Data Providers. These extensions are compiled as .dll's and included in the appropriate folders inside the root directory for the application, under the control of an extension manager.

1. **Experiment Types** are specific to one type of data analysis activity. For example, "MS" experiment type in the HDX analysis package uses a grouping of components to analyze and compute HDX for regular LC-MS data.
2. **UI Modules** are the main interaction interfaces associated with particular experiment types. Using AvalonDock, the user interface is set up so that all UI modules can be docked dynamically into the pre-defined regions of the UI shell. The new UI modules are created and dynamically loaded via Prism and MEF.
3. **Processes** are responsible for performing specific data analysis routines. They consist of multiple reusable algorithms (e.g. peak detection). Processes are loaded dynamically and controlled via a common interface for accessing the various algorithms and their corresponding parameters.
4. **Data Providers** are the components responsible for loading data into Mass Spec Studio. As a default, we include the MS file reader from Thermo and Proteowizard [1]. Proteowizard can read a large number of common mass spec data formats, and is an open-source library (see http://proteowizard.sourceforge.net/). We also support custom data providers.

Discovery, linkage and communication across standalone components, as well as with the main user interface, is achieved using the following agents:

1. A **Service Locator** retrieves any installed component.
2. An **Event Aggregator** sends and receives events between modules. It is the main channel of communication between independent modules, and ensures proper encapsulation and minimizes inter-dependencies.
3. A **Region Manager** links the main user interface with the modules.
4. A **Document Manager** links an experiment with the functionality presented within the main interface. Commands such as Open/Close are relayed through the Document Manager to the appropriate dependent modules.

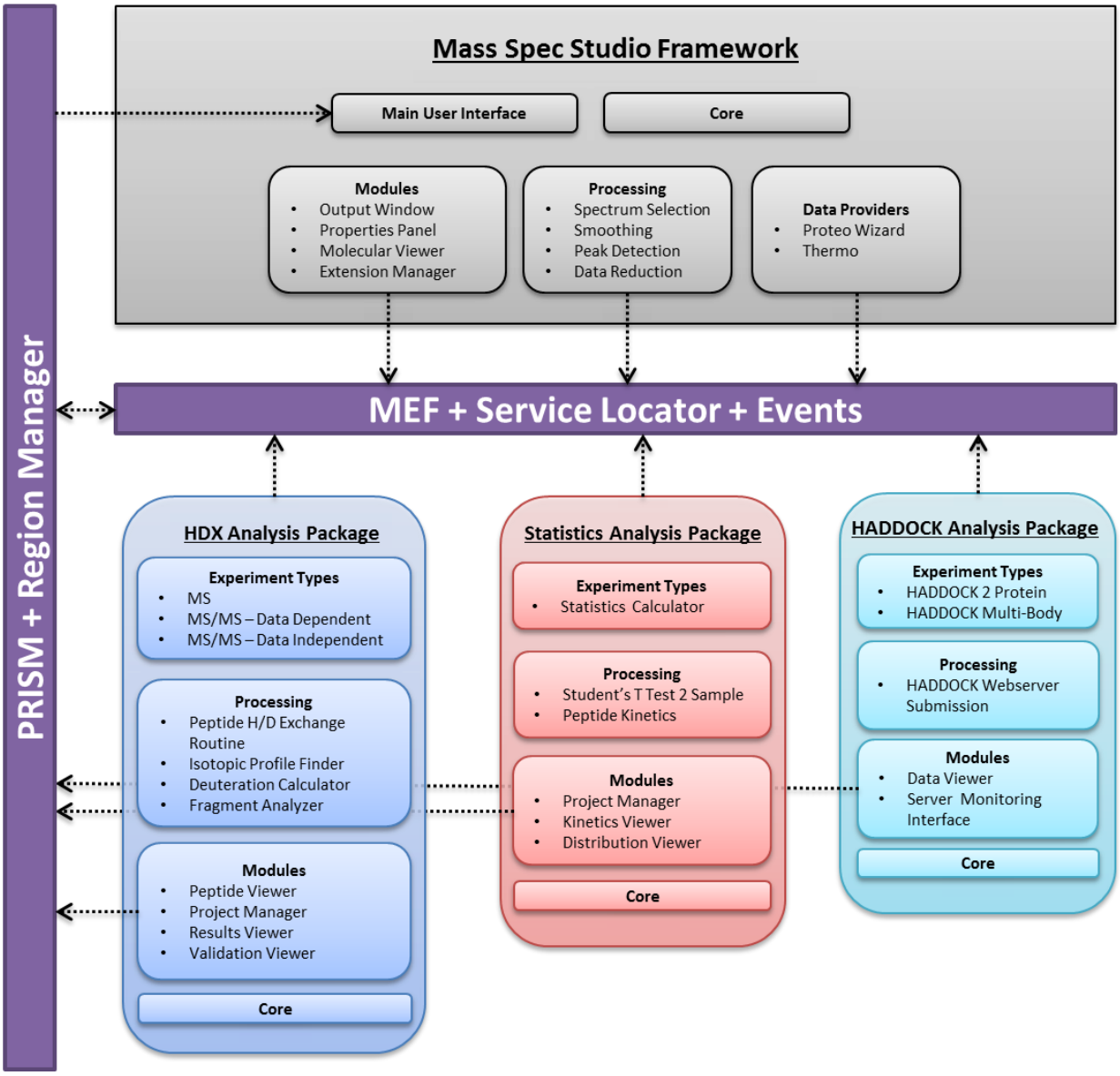


Figure S1. The schematic highlights a representative set of analysis packages available with Mass Spec Studio, and is a more comprehensive version of Figure 1 from the main paper. The communication layer consists of MEF, Service Location and global events propagation. The UI controller consists of PRISM with the Avalon Dock Region Manager. UI modules register with the UI controller, while any other discoverable components register with the commination layer.

The analysis packages currently available in Mass Spec Studio are HDX, covalent labeling, statistics, and restraint-based structure modeling. The packages share a simple communication protocol that allows data to be easily exported from one analysis package and exported in another. The main format for package data exchange is .CSV so that we do not enforce the Studio as a start-to-finish workflow. While our packages offer a variety of useful tools for the user to establish an efficient integrative MS workflow, we recognize that a user may want to merge with their own procedures, or insert modeling data from other sources (**Figure S2**). For example, the user may simply want to exploit the fast processing and validation functionality of the HDX package, and then export to a custom statistics procedure. The statistical implementation in the Studio can be bypassed. Similarly, other restraint data can be brought into the modeling application, such as arising from cross-linking. Ultimately, our data exchange feature allows the user to guide the workflow in the most efficient way possible for an individual use case.

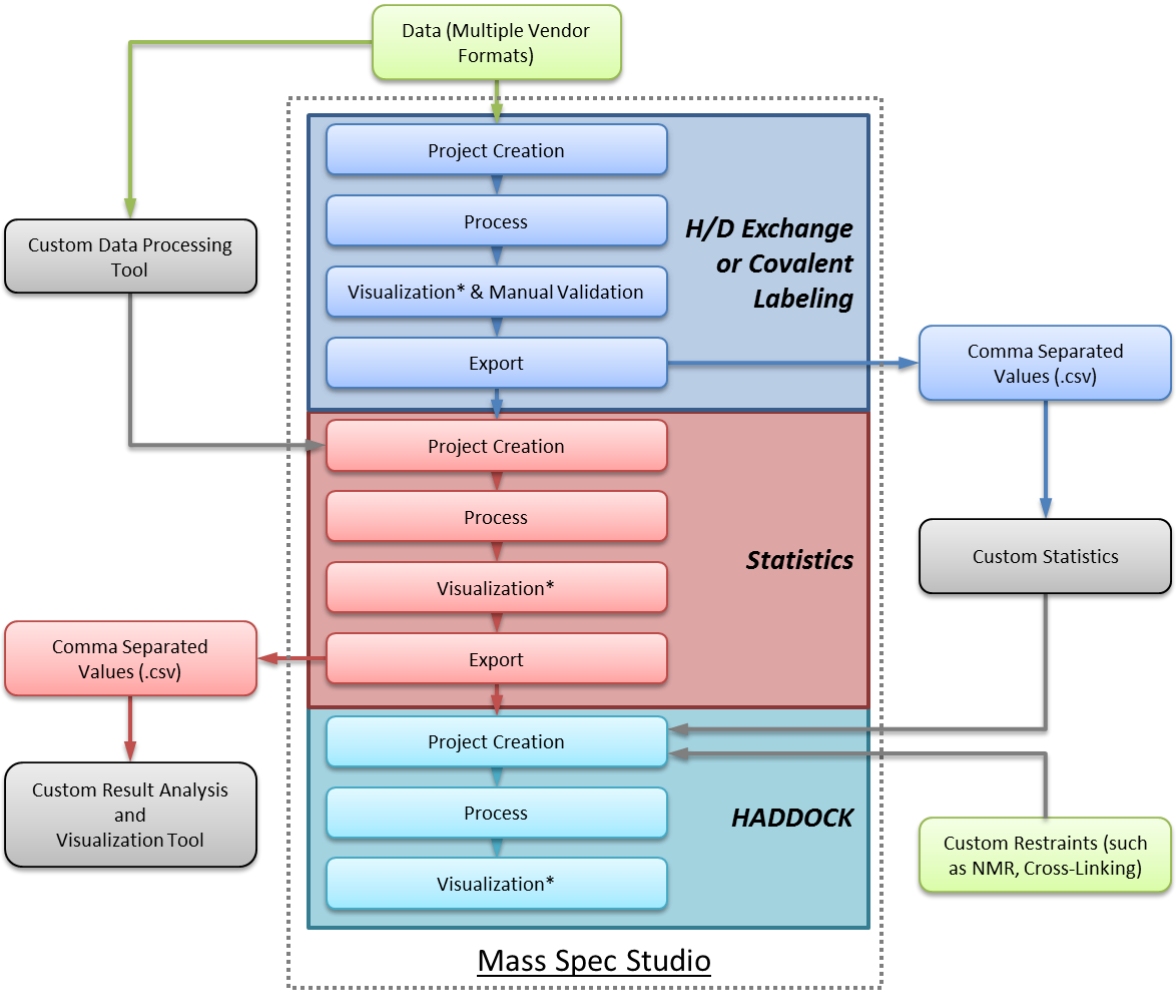


Figure S2. The analysis packages within the Studio communicate via a common CSV format. This provides at all times a common human readable format, which can be used to fully customize a data analysis workflow. Specifically, the user may swap any given analysis package in the Studio with any un-related analysis tool.

HDX Analysis Package

This package contains tools to analyze and visualize H/D data, built upon previous Hydra HDX software [2]. The user can import mass spec data, process data using a selection of algorithms and visualize the results. The multicore-enabled architecture of the HDX package can make full use of the machine's computing resources, and offers an efficient workflow for validating the results. The package offers several experiment types that can be utilized to make full use of different methods for monitoring HDX.

The ***1D HX-MS*** experiment type allows the user to supply MS data, as well as a list of known peptides. It is the default experiment, where the peptide feature map is known in advance, for example from a Sequest or Mascot database search. The processing routine to be used for this experiment type is "Peptide HX-MS", which computes deuteration values for each peptide in the feature map, based on centroid values and/or distributions (**Figure S3**).

The ***Targeted HX-MS^2*** experiment provides a fragment-level extraction of labeling data, and uses a similar project-building schema. The experiment is designed to support a data-dependent workflow or targeted acquisitions of MS/MS data, which usually incorporates a survey scan with each cycle. The "Peptide HX-MS/MS" routine can compute deuteration values for each fragment, in addition to the deuteration value of the associated intact peptide. This experiment types supports both collision-induced dissociation (CID) and electron-transfer dissociation (ETD) as methods for peptide fragmentation (**Figure S4**).

The ***HX-MS^2 experiment*** type is designed to support data-independent acquisition (DIA) of deuteration data, in both the peptide and fragment space. It currently supports AB Sciex SWATH™ acquisition concepts [3], however any DIA implementation scheme can be handled. Here, the acquisition must be broadly structured as a DDE experiment, except a series of m/z ranges should be employed for ion selection and transmission into the fragmentation event region, rather than discrete m/z values for selected peptides. Some elements of the data analysis procedures are considerably different than the previous experiment types (**Figure S5**). The experiment is configured in a straightforward manner, following similar workflow design concepts of the other experiment types. After visualization and validation, results can be exported in CSV format as with the 1D method, however the files are necessarily more complex. An export wizard allows the user to choose data fields (headers) to export. By default, the exported CSV file will contain the list of peptides first, followed by all of the fragments. The correlation between a given peptide and a fragment in the CSV file can be made by linking ID numbers. Due to the high computational cost of high resolution DIA experiments, we currently process and validate one run per project. Processing times scale with protein size and spectral complexity. An i7 Quad Core Windows 7 x64 with 8GB of RAM computer can process a 200-residue protein in 10 minutes, assuming a worst-case peptide library (i.e. no cleavage rules, so that any is peptide possible). This increases to 45 minutes for a 500-residue protein.

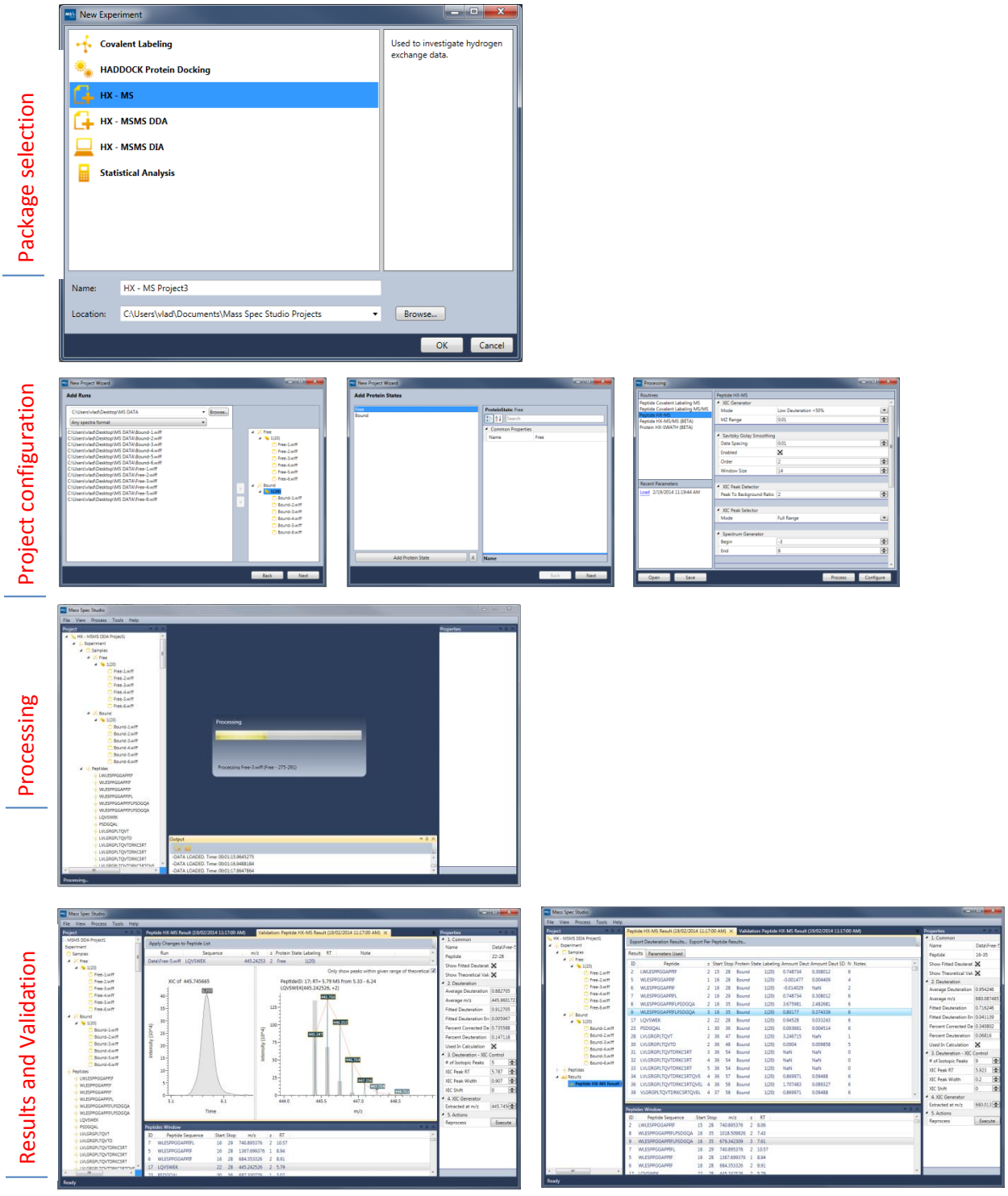


Figure S3. The user selects the HX-MS experiment type, and then builds a project tree containing all the replicate LC-MS files necessary for the analysis, and an input peptide feature list. Associations are established (e.g. labeling times, protein states) and then the data is processed after a selection of processing algorithms are suitably parameterized (e.g. peak detection, smoothing). Results are available in tabular form. They can be queried and revised using an efficient validation routine.

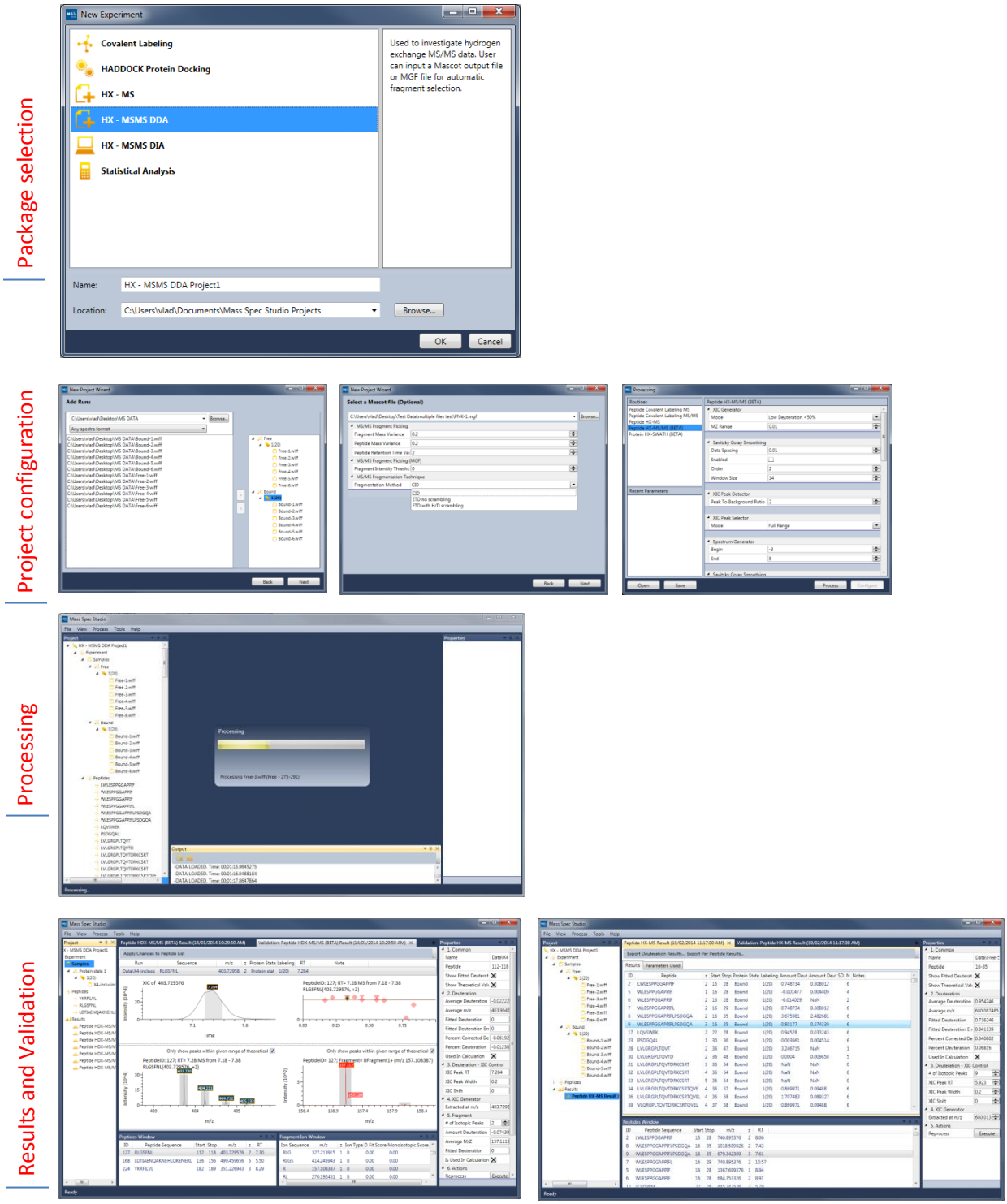


Figure S4. After building the project in a fashion similar to the 1D HX-MS experiment, the user processes with the "Peptide-HX-MS/MS" routine in order to calculate fragment deuteration. In addition to the MS data, there is a table of fragments associated with a given peptide, and the validation display contains two additional graph views. One display is to support validation of the MS/MS data for each fragment, and one is for plotting each fragment deuteration value against the expected deuteration value, for a given peptide.

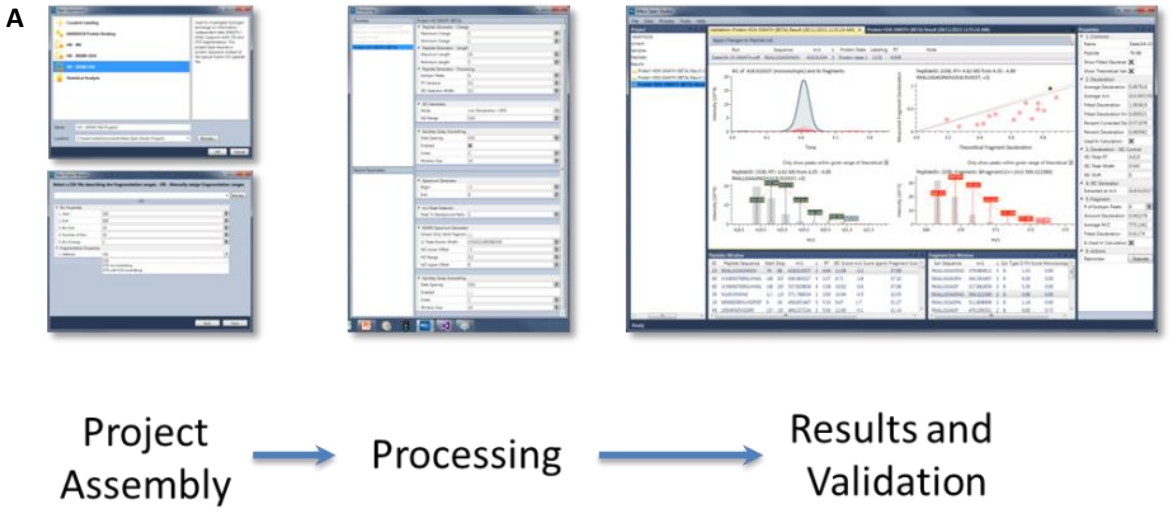


Figure S5. (A)The user defines a new project by selecting the DIA option, and associates the data files as per normal. The DIA transmission window sizes and locations are specified, as well as the intact protein sequence(s), and there is an option to include a list of expected peptides instead. A more involved processing window allows for data analysis parameterization in both MS and MS/MS space. After processing, the validation viewer allows the user to mine and adjust the processing in both dimensions. (B) The associations between fragments and precursor ions are built using a peptide library. The library is assembled using prior knowledge of the protein sequence(s) in the sample, and the enzyme specificity, if known (if not, all peptides are considered). For each library member, a search is conducted for evidence of existence, using XICs generated in both MS and MS/MS space. Alignment between peptide and fragment XICs is determined, where the XIC search in MS/MS space is based on expected sequence ions (e.g. b and y series for CID). Alignments are scored using the X!Tandem scoring algorithm [4] and identification of the peptide is based on a weighting of this score and the accurate mass of the peptide. In situations of conflict, that is, if fragment XICs align with two or more peptide XICs at the same retention time, they are not used in scoring either peptide, nor in the extraction of deuteration data. This ensures only unique fragments are considered. Deuteration data is computed for both MS and unique MS/MS fragments, and exported in the master CSV file.

Covalent Labeling Analysis Package:

The covalent labeling experiment type is contains a package of tools adapted for the analysis of labels arising from chemically or photolytically-induced oxidation, reactive functionalities (e.g. amine-specific succinimides, carboxyl-specific carbodiimides, broadly reactive carbenes). Combinations of covalent labeling can also be accommodated. The workflow is conceptually similar to HD methods, in that labeled peptides need to be identified, and the distribution of incorporation determined. However, analysis is complicated by signal splitting in the LC space, as label insertion locations can alter chromatographic properties. The workflow supports both MS and MS/MS based analyses, for label quantitation in both dimensions (F**igure S6**).

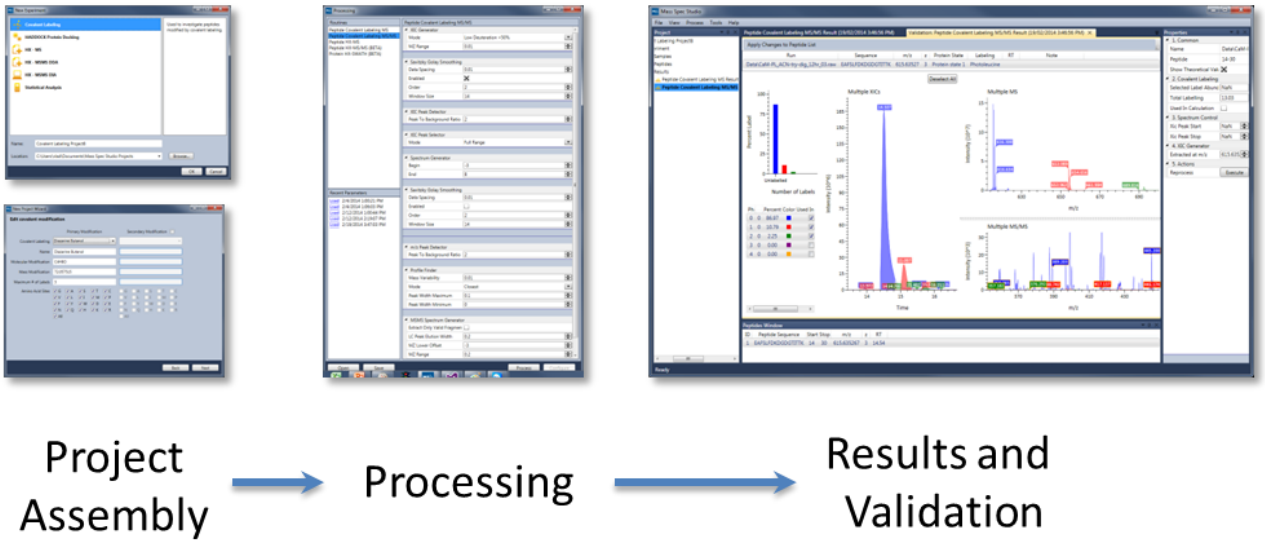
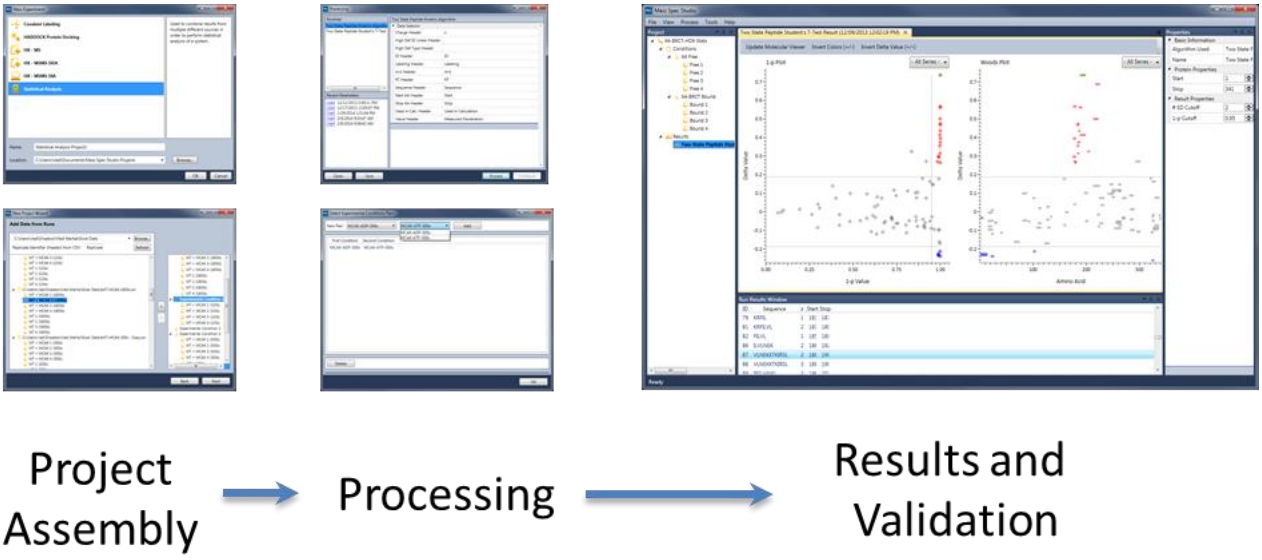


Figure S6. After selecting the covalent labeling experiment type and linking the project files, the mass shift associated with the labeling chemistries is determined from the molecular formula provided by the user, and data processing is configured. Both LC-MS and LC-MS/MS data can be processed. The workflow is most similar to that of the High Resolution HX-MS. Label quantitation is determined in MS data from the accurate mass of the labeling events and can be supported by fragmentation analysis if this is acquired in the run. Label quantitation can be visualized and validated in an interactive manner, and the results can be quantified for unique features or as a distribution across all chromatographically distinct features.

Statistics and Structural Visualization analysis package

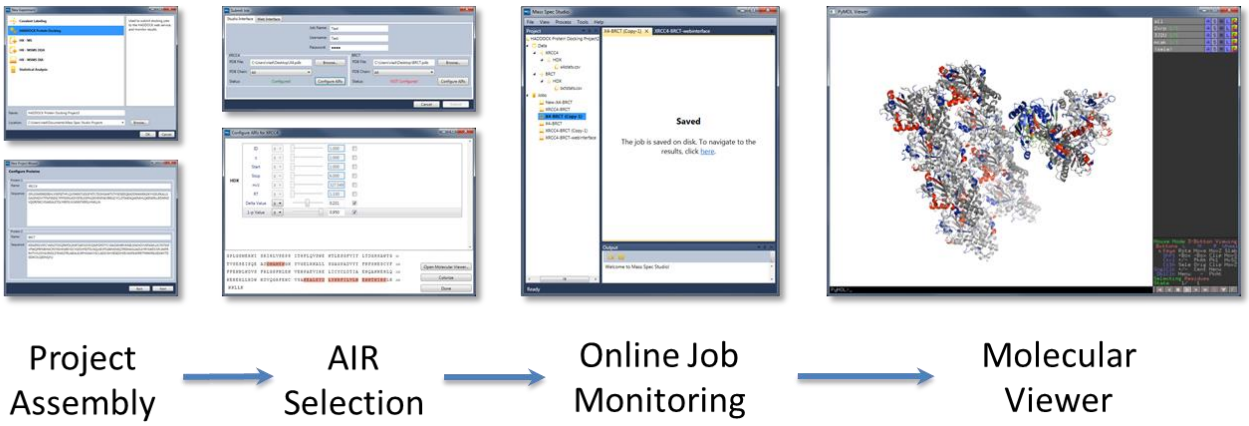
The statistics package accepts the output of any processed and validated data analysis routine, whether HDX or covalent labeling experiment types, and promotes an interactive analysis between the processed data and available structural representations of the protein(s) being analyzed. The current views interact with data processed in a typical pairwise fashion, although there is no restriction to such. That is, an experiment in which multiple types of perturbations are applied – for example a selection of different drugs, or different binding partners in a multiprotein complex – can be accommodated. The output is fully interactive, so that peptides of interest can be viewed within the context of a molecular viewer (**Figure S7**). Simple pairwise statistics are utilized here, although other methods can be selected. The statistical output of deuterium label incorporation can be viewed also as kinetics plots (not shown).



Supplementary Figure 7: The statistics analysis package is selected, and then a series of processed results files are associated to the new project, for statistical and visual evaluation. Here, statistically significant mass shifts are determined by calculating p-values from the replicate data on a per-peptide basis, using a two-tailed Student's t test using pooled standard deviations, and by requiring shifts to exceed a threshold value (±2s.d.) based on a measurement of the shift noise. This analysis generates a 1-p plot, and is paired with a Woods plot to align the statistical treatment with position in primary structure terms. The plots are interactive, such that selecting a peptide in either plot highlights the corresponding sequence in structure, using PyMOL as the molecular viewer.

Molecular modeling

Restraint-driven assembly of higher order structure, particularly for protein interactions, routinely generates structural representations of high accuracy. We incorporate HADDOCK as one of the most flexible utilities for such purposes[5]. The HADDOCK package in the Studio provides a direct interface between the MS data processing and the HADDOCK web service. The Studio automates the data pre-processing and requires minimal supervision from the user (**Figure S8**). Additionally, several tools allow interaction with the resulting HADDOCK structures.

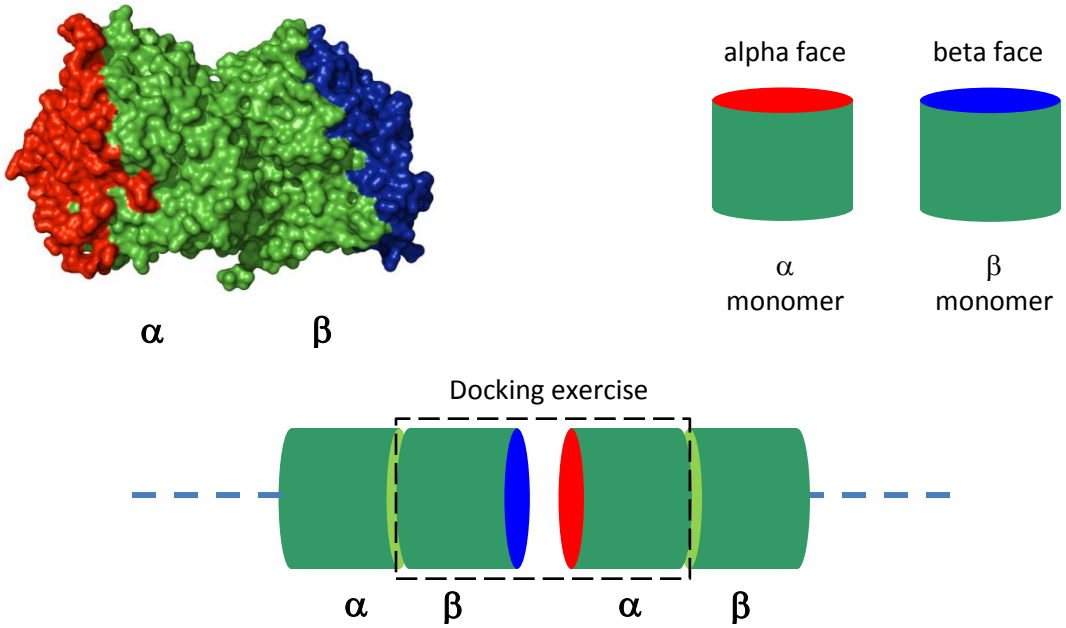


Supplementary Figure 8: The structure modeling concept using MS data is based on the determination of ambiguous interaction restraints (AIRs) extracted from active residues, from the statistical output generating in processing steps. The user selects the output of the statistical analysis and then applies a customizable selection strategy to the perturbations, which generates a set of AIRs. AIRs are uploaded to the HADDOCK server and submitted as a docking job. The job process can be tracked via the Mass Spec Studio interface as well as through the regular HADDOCK web interface. Finally, the molecular viewer interface may be used to visualize the docking solution clusters.

A method of AIR generation for HADDOCK

There is extensive literature on the operational aspects of HADDOCK, and the nature of the output. Here, we highlight aspects of the analysis that are unique to our docking examples and the processing of chemical labeling data, and restrict our treatment to conventional labeling. Conventional labeling data is acquired at the peptide level, and in most cases map to sections of structure that are larger than the actual impact of a binding event (low structural resolution). Together with partial sequence coverage, noise in the label measurements, and distal effects of binding, interfaces are fuzzily defined and buried amongst other data. Our goal is to determine the subset of label data that generates an optimal dock, one that can be tested by independent measures.

We illustrate one strategy to achieve optimization using HADDOCK docking output, by simulating the assembly of tubulin dimers into tubulin protofilaments. This involves a reduction of the assembly process into the docking of two monomers, α and β, from two different dimers (**Figure S9**).

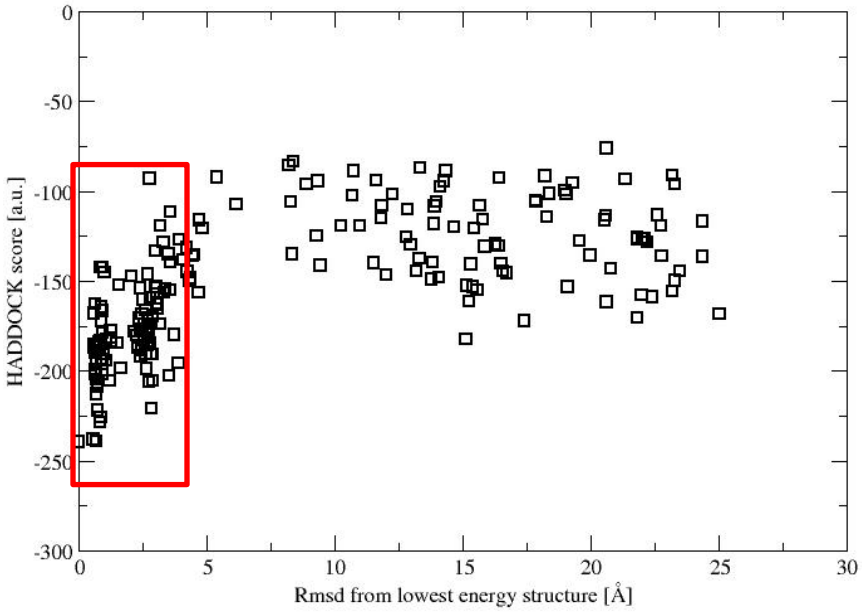


Supplementary Figure 9: A redocking exercise. The interfacial region created upon tubulin polymerization. Docking in HADDOCK consisted of assembling α-tubulin from one dimer with β-tubulin from a second dimer, using active residues selected from the known interfaces from PDB accession code 1JFF. Any residue within 5Å of the other unit across the inter-dimer interface and possessing a relative Solvent Accessible Surface Area (rSASA) greater than 50% was designated as an active residue. Passive residues were defined automatically based on >50% rSASA and within 5Å of any active residue in the set.

Here, we used known interfacial data for a redocking exercise, conducted on a local installation of the HADDOCK2.0 scripts and the CNS simulation engine. Setting up the docking required some additional processing of the protein structures and the bound ligands, particularly GDP bound at the interface between α and β tubulin. First, to avoid problems with the force-field description of a gap in a peptide chain, all HADDOCK simulations were performed with 1JFF structures treated with MODELLER to fill the chain gaps using homology modeling based on PDB accession code 1Z2B. Second, proteins were

modeled using the force-field parameters of Linge et al. with the non-bonded parameters of OPLS. Third, the GDP nucleotide was modeled with a modified set of GTP parameters originally from the PRODRG server. The nucleotide was included in the β unit file in the relative position indicated in the 1JFF structure, and its motions constrained using supplemental restraints added to the HADDOCK AIR file . All other molecules such as ions and organic compounds were removed before input to HADDOCK. Finally, the 1JFF heterodimer PDB file was split into two separate files, one for the α unit and one for the β unit, as input for HADDOCK.

The HADDOCK protocol begins with the "it0" stage, where each protein is held internally rigid. The two proteins are separated and randomly oriented before being allowed to move together. When water docking is used, the it0 stage also includes the addition of a water layer about each protein before they are "docked" together. This water layer is partially removed through diffusion and a Monte Carlo based removal of less favorable water molecules. For this test, 2000 rigid docked structures were formed. The it0 structures were then analyzed and sorted by the HADDOCK score. The 200 best-ranked structures were forwarded to the "it1" stage. In it1, the docked structure is optimized through an angle-dihedral space simulated annealing procedure, which is followed by a short MD relaxation in explicit solvent. The resulting 200 structures were then analyzed, sorted by HADDOCK score, and RMSd clustered with a clustering radius of 4 Å. The best HADDOCK score representatives of each cluster were visually inspected. The McLachlan algorithm, implemented in ProFit, and procedures provided on the HADDOCK website were used to RMSd cluster resultant structures from each simulation (**Figure S10**).



Supplementary Figure 10: The RMSd clustering results of the protofilament derived AIR simulation. A best group size of 82 was returned as the two semi-distinct groups below 4Å RMSd.

The redocking example returned 82 representative structures that differed very little from the 1JFF original structure – within 4 Å RMSd. The remaining samplings were diffusely spaced or weakly

clustered. Therefore, we chose a strategy for implementing labeling data based on the optimization of cluster size, and only loosely on HADDOCK score.

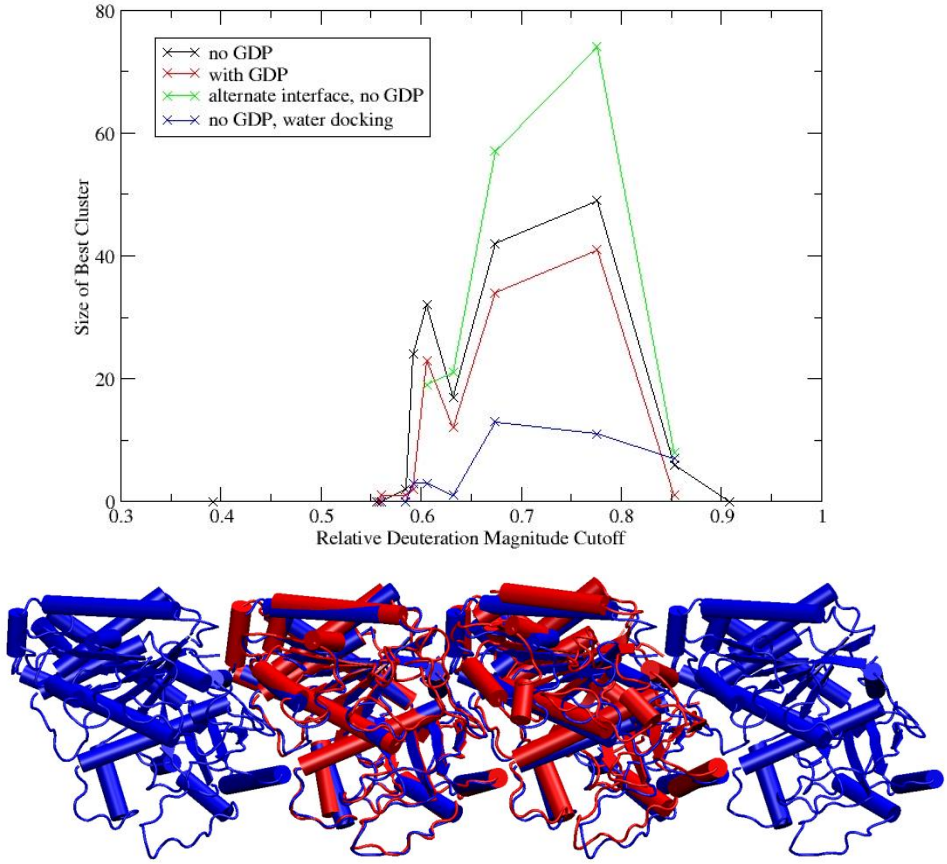
First, we produced HX-MS data from free tubulin dimer and assembled tubulin, and rank ordered the peptides according to the degree of change in deuterium labeling. In this example, we restrict the assessment to peptides in the general vicinity of the known interfaces (i.e. excluding known lateral contact regions between dimers). The HX-MS experiments presented 17 peptides intersecting the dimer interface, with lengths ranging from 6 to 19 residues, for a total of 182 residues (Supplementary Table 1). We then used the magnitudes of altered deuteration for assembling active and passive residue subsets and driving the simulations. A series of subsets were defined, starting with a minimal set of peptides with the 0-0.4 deuteration ratio, with successive subsets containing the peptide with the next higher deuteration ratio in the list. This generated a total of twelve AIR subsets defined on the basis of deuteration ratio cutoffs. Active residues were defined as any residue in the peptides with >50% rSASA and passive residues were defined as any residue within 5 Å of an active residue, and >50% rSASA. The largest residue set contained 37 active residues and 14 passive residues.

Supplementary Table 1. Subset of peptides with significant alteration in deuterium levels resulting from tubulin polymerization, encompassing all peptides intersecting a crude estimate of the interdimer interface.

Start-stop[a]	Sequence	MT/Dimer[b]
β181-189	T.VVEPYNATL.S	0.09
α344-351	F.VDWCPTGF.K	0.16
β399-407	M.FRRKAFLHW.Y	0.21
β73-80	P.GTMDSVRS.G	0.27
α253-271	L.TEFQTNLVPYPRIHFPLAT.Y	0.39
β103-114	N.WAKGHYTEGAEL.V	0.56
β68-75	L.VDLEPGTM.D	0.56
α319-330	L.YRGDVVPKDVNA.A	0.56
β169-180	T.FSVVPSPKVSDT.V	0.59
α243-248	L.RFDGAL.N	0.59
α352-368	F.KVGINYQPPTVVPGGDL.A	0.61
β93-102	F.VFGQSGAGNN.W	0.63
α120-134	L.DRIRKLADQCTGLQG.F	0.67
α260-271	L.VPYPRIHFPLAT.Y	0.71
β215-232	F.RTLKLTTPTYGDLNHLVS.A	0.78
α335-343	T.IKTKRSIQF.V	0.85
α158-169	L.SVDYGKKSKLEF.S	0.91

[a] sequence position in α and β tubulin
[b] ratio of deuteration levels for the assembled (MT) and the dimer states.

Each subset of restraints were translated into AIRs within HADDOCK and the results generated as described for the redocking experiment. Clusters (4 Å RMSd) with scores in the general range of the redocking exercise were tracked as a function the deuteration ratio cut-off. The lowest energy cluster (based on HADDOCK score) is represented in **Figure S11**, and demonstrates that cluster size moves through an maximum as a function of the deuteration cut-off.



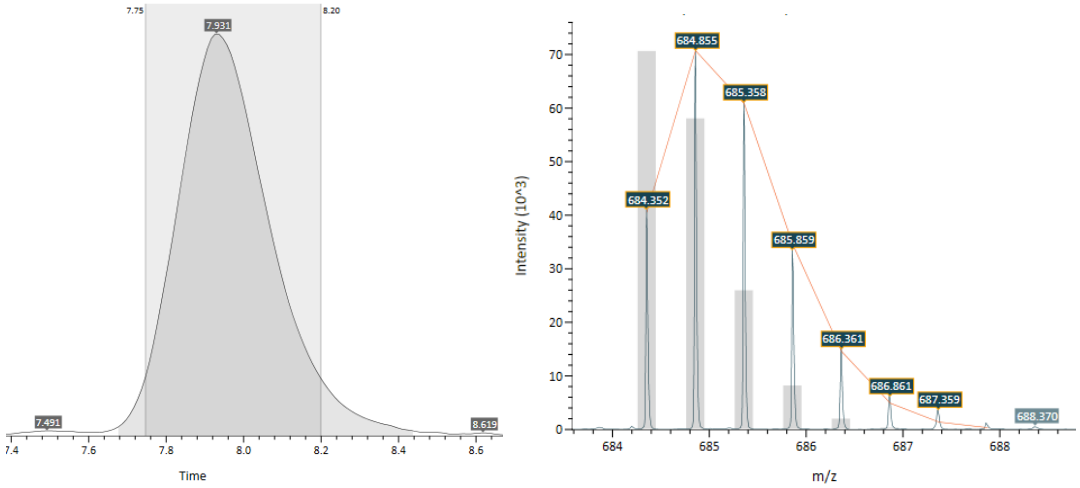
Supplementary Figure 11: The size of the lowest energy cluster was tracked as a function of deuteration cut-off, for a series of related HX-MS based HADDOCK simulations. The "alternative interface, no GDP" represents the inclusion of residues well outside of the known interface. Regardless of the configuration of the dock (with or without nucleotide or interface water), the cluster size exhibits a maximum at 0.78. The best control result was a cluster size of 82 (see **Figure S10**). Included is the RMSd-fitted superposition of a representative structure (red) from the lowest-energy cluster of the 0.78 deuteration ratio cut off simulation, onto the 1JFF protofilament model (blue). The HADDOCK result and the protofilament model share nearly the exact same inter-dimer interface.

These results shows that optimizing cluster size using a strategy of iterative AIR generation can be successful in generating testable high resolution models of interactions. Although an iterative strategy involved greater computational time, it provides a measure of robustness and helps minimize selection of docked models based only on score, which can be error prone.

Interactive Graph Control

The interactive graph control provides a new level of usability for visualizing and validating MS data. Unlike other general graph controllers such as Zedgraph, our graph control was developed and optimized specifically for mass-spectrometry applications. It takes full advantage of data caching and hardware acceleration provided by the Windows Presentation Framework (WPF), giving it the ability to render very large amounts of data points in real-time.

The graph control is built as a pluggable module into the Studio framework and can be re-used by any content developer down the road. To enable this, many properties and action triggers are exposed, which allows a high degree of customizability.



Supplementary Figure 12: On the left, an extracted ion chromatogram with a click-and-drag selection event to specify a new range for spectrum selection. On the right, the resulting spectrum with the highlighted peaks (darker) manually selected for inclusion in the mass shift calculation.

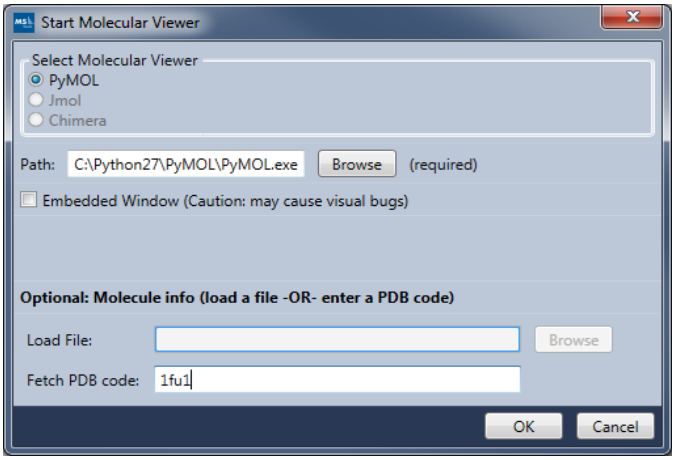
The user interface of the graph control is highly interactive. One can trigger various actions by clicking on peak labels, dragging across the axis, or even dragging across the graph content. Because of this, cluttering UI elements such as peak lists and other tables become less necessary, as all the required components can be visualized and manipulated directly using the graph control.

We highlight some of the strengths of the graph control in the HX analysis package. During validation of a HX-MS project, a user is presented with a minimal view of two graphs, an XIC and a spectrum (**Figure S12**). The user may control how the averaged spectrum is extracted by re-selecting an area on the XIC graph. For deuteration calculations, a user may change the default peak selection to correct any mis-selected peaks from the automatic analysis.

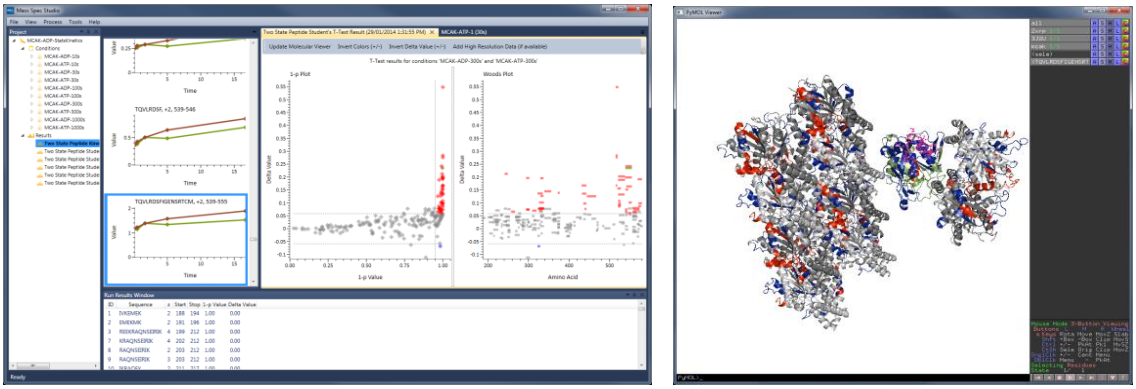
Molecular Viewer Tool

The molecular viewer interface is a package-independent module offered with the Mass Spec Studio library, which can be invoked in any application using peptide data. It offers a communication channel for external components to control the molecular viewer (**Figure S13**). To use this tool, a user must have a molecular viewer installed on their system. We currently support PyMOL. The HX package, Statistics package and HADDOCK package have established links to the molecular viewer interface. A user may open their protein of interest via the molecular viewer interface, and any linked UI components (such as peptide tables) can control the molecular viewer. For example, selecting a peptide will highlight that sequence on the overall protein structure. In statistics, the user may colorize the protein by clicking "Update Molecular Viewer" on a given statistical analysis result. This action will apply the colors associated with the statistical significant changes from the peptides onto the 3D molecular structure. Using the molecular viewer via the Mass Spec Studio interface does not restrict its functionality relative to using it as a standalone application.

A



B



Supplementary Figure 13: (A) The command window for controlling the molecular viewer interface. This can be accessed from navigating to the main menu, hovering over "Tools" and selecting "Molecular Viewer". If the path field is empty, a molecular viewer could not be automatically found. If there is one installed on the system, the user must manually find it's ".exe" via the "Browse" button. (B) Selecting a peptide in the kinetics view or in the peptide table will highlight the respective sequence on the molecular viewer window (purple dots).

Tutorial: Project creation and processing HX-MS files in the Studio

Project creation wizard

Begin by opening MS studio, and creating a new experiment under the "File" tab. Select "HX-MS" as the appropriate experiment type for LC-MS data analysis (**Figure T1**).

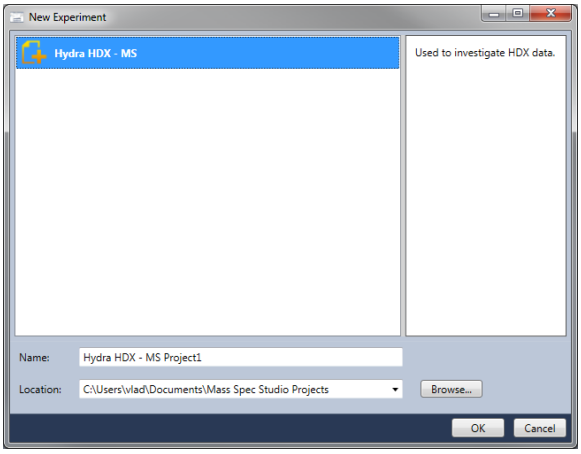


Figure T1. Selecting the experiment type to use from the Mass Spec Studio.

Fill in a suitably descriptive name for the Project. Specify where to save the project using the "location" field. By default, Studio projects are saved in "C:\Users\UserName\Documents\Mass Spec Studio Projects". Click "OK" to continue. A new window will pop up, which allows you to create the Project Tree, which establishes the relationship between protein states, labeling conditions and the assortment of data files created in the lab (**Figure T2**). Click "Add Protein State" twice, and name each one appropriately, to establish the two branches in the Project Tree (for example, bound state and free state). To change the name, highlight the protein state name in the protein state window and type the new name in the "Name" field, in "Common Properties" in the right window pane. Click "Next" to continue. In the "Add Labelling" window, specify the different labelling conditions used in the experiments. A time-course at a given $\%D_2O$ can be established here. Click "Add Labelling", and enter the $\%D_2O$ in the "Percent" field and the new labelling time in the "Time". Both are found the Common Properties pane on the right of the window (**Figure 2**). Users can delete any entry in this tree definition exercise, by highlighting the item (Protein State or Labeling) and selecting the "X" at the bottom of the pane.

Click "Next" to continue. The Data Provider can be changed using the "Data Provider" rolling menu. The Studio framework can support certain vendors natively (.raw data from Thermo Fisher and .wiff data from AB Sciex) whereas other vendors can be accommodated through ProteoWizard. ProteoWizard offers access to most MS vendor file formats, but the reader is the only feature of the ProteoWizard Library and Tools set that we use. Internally, the Studio converts the file into a unique format, for processing efficiency. Click "Next" to continue. You can now distribute the data files throughout the Project Tree that you have created (**Figure T3**).

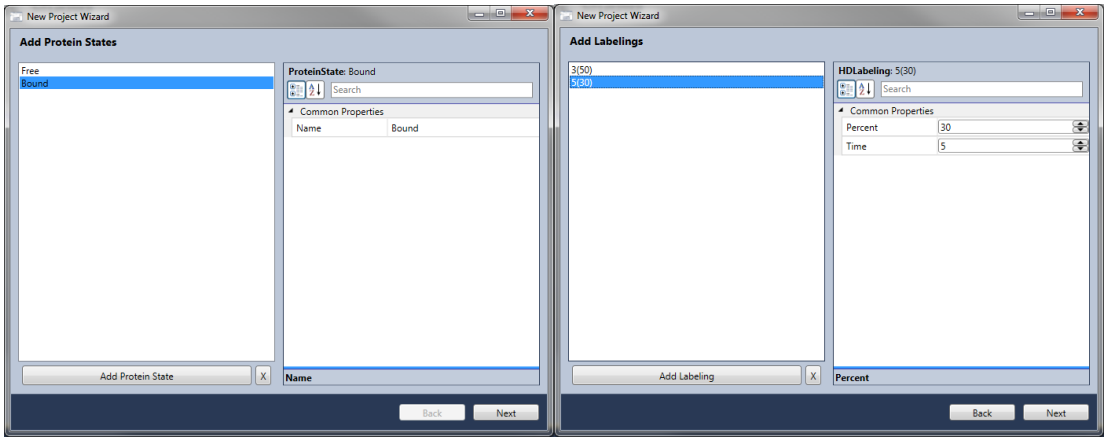


Figure T2. Creating the Project Tree by defining protein states, labeling conditions and labeling times.

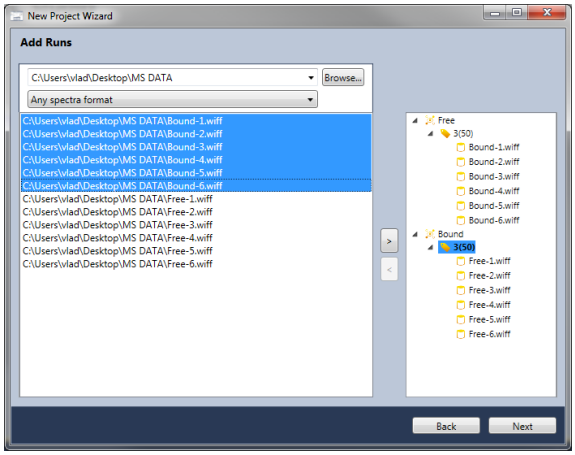


Figure T3. Distributing the files in the Project Tree.

In the “Add Runs” window that pops up, navigate to the folder containing your data, and the list will populate with the data files present in the folder. Highlight one (or several) files and then highlight the destination branch in the tree, shown on the right side of the pane. Link the files to the project tree by selecting the right-pointing arrow in the middle of the pane. Errors can be corrected by highlighting the linked files and selecting the left-pointing arrow.

The next window in the wizard allows you to link a peptide list to the project (“Import peptides”). Click “Browse” to find the folder containing the .csv file with the necessary peptide information. An example peptide file is provided, as a template, with the software download package. The headers in the .csv file that are recognized by the software are listed below. The headers that are underlined are required, while the rest are optional.

- **ID** – The unique identifier of the peptide (0 - n).
- **Protein** – The protein or protein mixture from which the peptide originates.
- **Start** – The start position of the peptide in the protein sequence (first amino acid in the protein is 1).

- **Stop** – The stop position of the peptide (inclusive). Example: GR**AFAYKL**SRK. The peptide "AFAYKL" would have Start = 3 and Stop = 8.
- **Sequence** – The amino acid sequence of the peptide. (Required)
- **z** – The charge of the peptide. (Required)
- **m/z** – The mono-isotopic mass to charge ratio for the peptide.
- **RT** – The retention time of the peptide.
- **RT Variance** – The value on either side of the RT where XIC peaks are considered for peak selection.
- **XIC m/z** – The m/z at which the XIC will be extracted (based on the mono-isotopic mass by default). The user may adjust this value in case the masses are shifted or a proper XIC is not found.
- **XIC Adjustment** – An RT value by which the XIC should be offset, from the original RT value.
- **XIC Width** – The retention time window over which the spectra will be summed.
- **Number of Isotopic Peaks** – The number of isotopic peaks used for calculating deuteration values.
- **Notes** – Any information about the peptide that may be useful to the user.

Once the peptide file is associated with the project, click "Finish". The Studio will now copy all the data associated with the project into a processing folder, and generate an internal compressed file format designed for fast XIC and spectrum retrieval. The time it takes for this step will depend on the computer used. After the generation of the cache, the main project window displays all the files and peptides for the project, in a hierarchical list on the left side of the window (**Figure T4**). Users can expand each section by clicking on the arrow to the left of any entry in the list. By default everything is hidden.

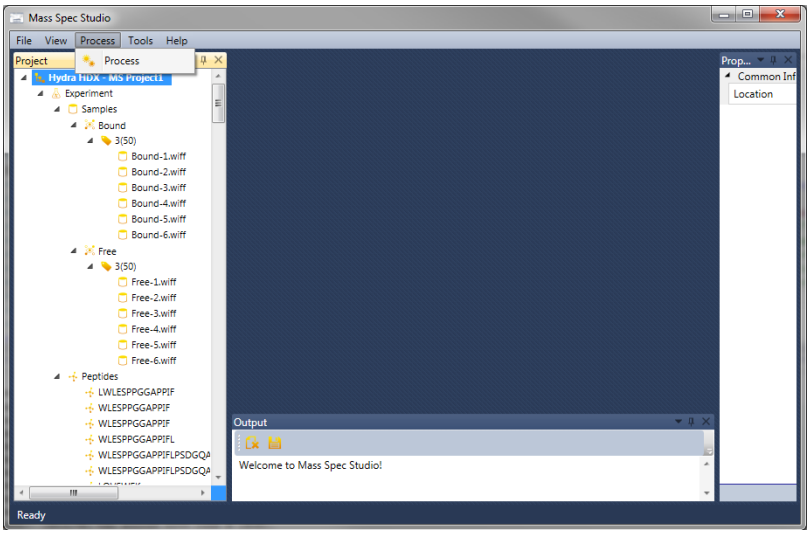


Figure T4. The assembled Project Tree, complete with data files and peptide sequences to be processed.

Processing

To begin the extraction of deuteration data, simply select “Process” from the main menu. The processing menu presents three panes (**Figure 5**). The “Routines” section in the top-left corner allows you to select a processing strategy appropriate for the experimental data you are using. The version we describe allows selection of “Peptide HX-MS”. The Peptide HX-MS routine generates average deuteration levels for each peptide uploaded to the Project Tree. Highlighting this routine refreshes the large right pane of the window, which presents a set of algorithms and associated parameters that are populated with default values. All the algorithms you would expect to see in an LC-MS data analysis are present, plus a set that are specific to HX data extraction. These include an “XIC Generator” to produce chromatographic traces for each peptide m/z in the list, “Savitzky Golay Smoothing” to aid in XIC peak detection, an “XIC Peak Detector” and “XIC Peak Selector” to find the peptides with the correct m/z. Once peptides are found, a mass spectrum is generated by integrating the selected XIC peak for a given peptide, using “Spectrum Generator”, and this can be smoothed with the “Spectrum Smoother” algorithm. The “m/z Peak Detector” locates all isotopic peaks in the spectrum and the “Profile Finder” identifies the peaks belonging to the isotopic cluster for the peptide of interest. The “Deuteration Calculator” calculates the average deuteration of the peptide from the isotopic cluster.

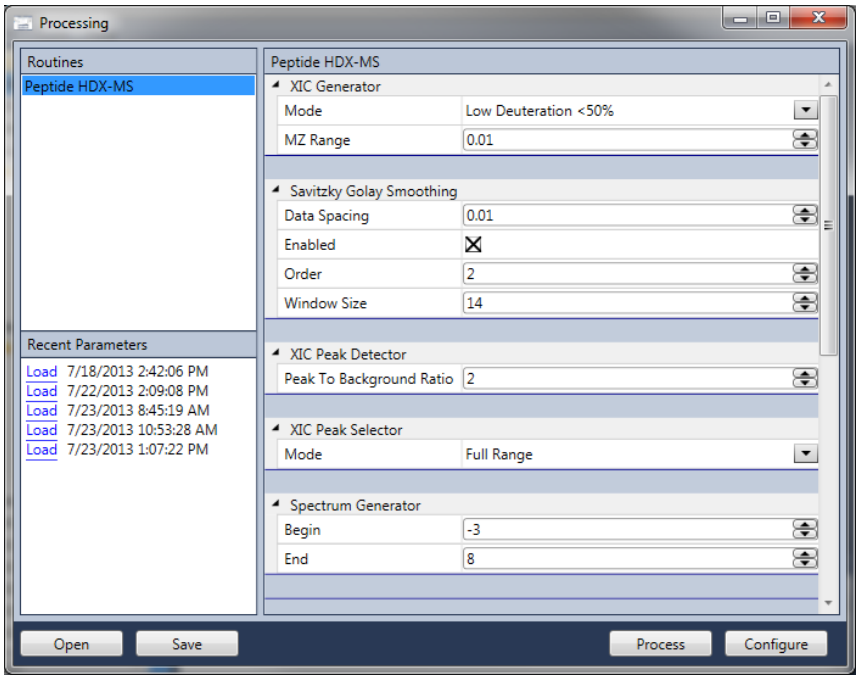


Figure T5. Processing window for configuring the algorithms used in extracting deuteration data from the Project files.

There are four ways you can proceed with parameterizing the algorithms. If you are a new user, we recommend selecting “Configure” (bottom-right) to launch a wizard. This wizard walks you through each algorithm in turn, and provides useful information to help you select the right parameters for your situation. Advanced users can change the parameters directly in the main window. You can rapidly recall

a set of parameters recently used, using entries in the bottom-left pane, or you can open a parameter file that was previously saved in an earlier project.

You are now ready to process your data - click the “Process” button on the bottom of the pane. During processing, the main window displays a progress bar, and a dynamically updating “output” window at the bottom (**Figure T6**). The speed of analysis is strongly dependent on the size of your Project Tree and the computer configuration, but you can expect rates of approximately 30-60 peptides per second.

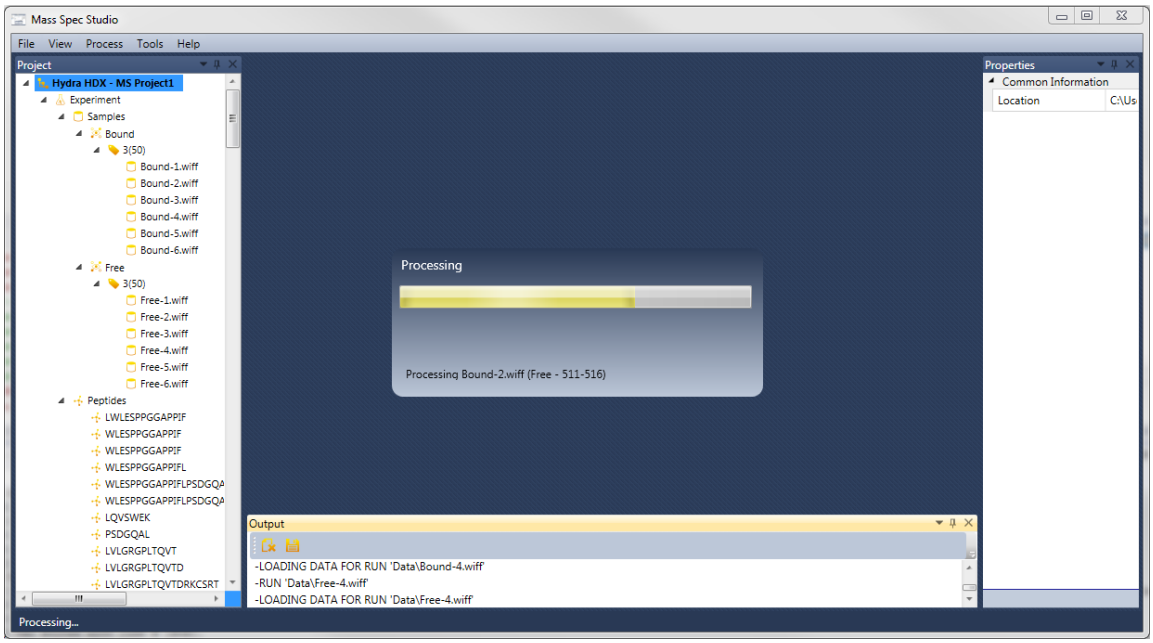


Figure T6. Active display during file processing.

When processing is complete, a report is displayed with a number of notable features. The display defaults to a “Validation” viewer, but the user can toggle between two tabs as shown at the top of the center pane (**Figure T7**). The “Peptide HX-MS Result” tab provides a summary table for each peptide in the top-middle pane, for each of the elements of the Project Tree. This global view offers a handy way of quickly evaluating the overall success of the analysis. For example, an average per-peptide deuteration value is generated using all replicates. You can export this summary table in a .csv format, or if you want the full set of data (i.e. un-averaged, with all technical replicates) you can export this as well. Just below the two main tabs you will find these export functions: “Export Deuteration Results…” for the averaged summary and “Export Per Peptide Results…” for the full set. We have included these export tools to support the use of other statistical packages, however the averaged summary contains basic statistics useful for many HX-MS applications (average, standard deviation and actual n values).

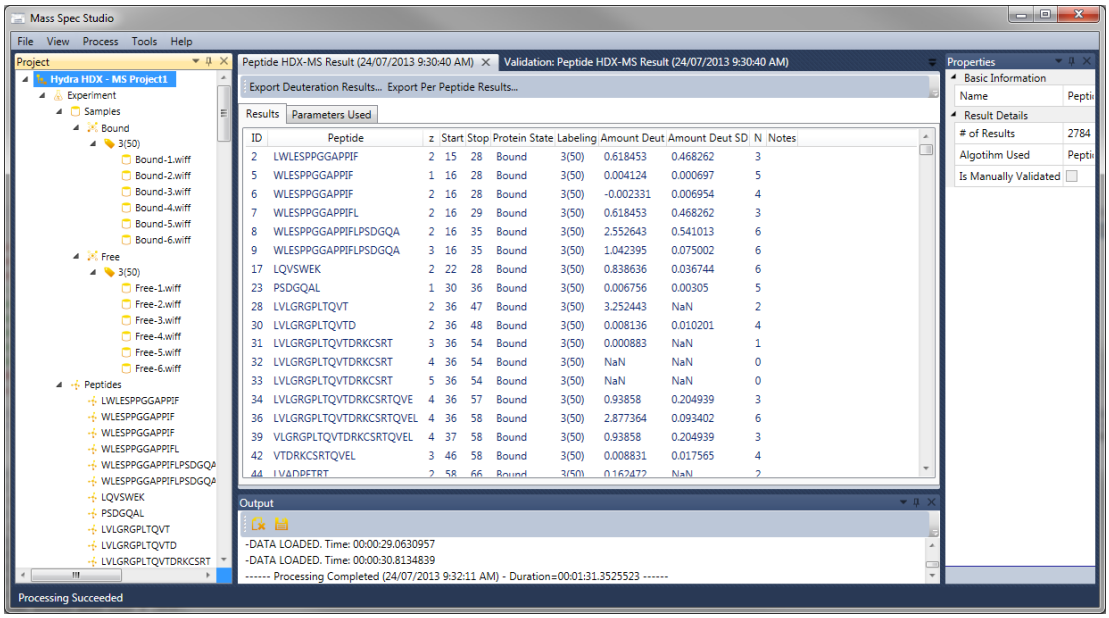


Figure T7. The results window, highlighting only the global results pane.

Visualization and Validation

Part of the philosophy behind the Studio design acknowledges the complexity in configuring the right combination and parameterization of data extraction algorithms, for any field of bioanalysis. We have designed an approach that allows the user to rapidly access the underlying raw HX-MS data, in order to visualize the impact of various parameters on data output. To access this mode from the results window, select the "Validation: Peptide HX-MS Result" tab at the very top of the center-top pane (**Figure T8**). It is not uncommon for a subset of peptide selections to present sub-optimal deuterium quantitation, because of partial peptide overlap or mis-selected chromatographic peaks. In this section of the software, you can adjust certain parameters for individual peptides in the list and then recalculate deuteration. Adjust settings using the ride-side pane called "Properties" as described in more detail below, and then select "Execute" at the bottom of this pane to reprocess. This refreshes the calculation for the peptide and updates the results summaries as well. The changes that you make can be saved to the peptide list by clicking "Apply Changes to Peptide List" on the top-left of the results window. This will update the peptide list in Mass Spec Studio for the project you are working on. To preserve this information for future projects, the user may right click on "Peptides" in the Project Tree and select "Export".

The results pane is organized to display the list of files being processed (top), a chromatogram and spectrum viewer along with peak statistics (middle) and the list of peptides being explored (bottom). Navigate the data by highlighting a data file on the top, and then changing the peptide selection at the bottom. This will refresh the viewers and the "Properties" window to the right of the viewers. The "Properties" window displays all the information on the peptide currently selected, and provides a few tools for adjusting data extraction from the peptide. There are several sections to this window.

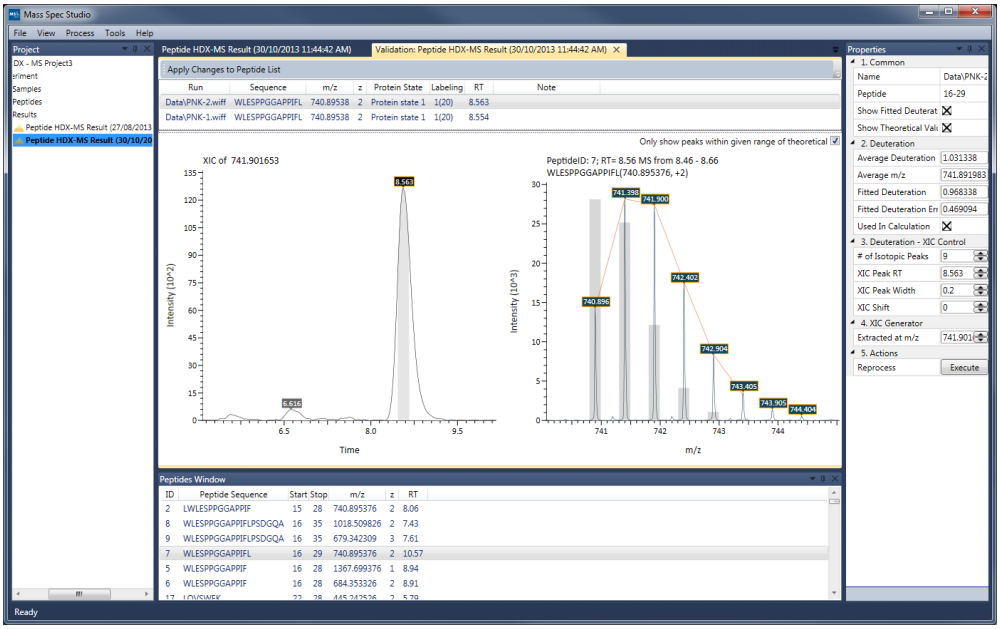


Figure T8. The main validation display, allowing the user to quickly view and adjust the extraction of deuteration data.

The “Common” part contains the name and the path of the data file, and the location of the peptide in the protein sequence. A check box allows you to activate or deactivate a display of the theoretical, non-deuterated, peptide isotopic distribution. This can be handy for a quick assessment of label uptake. The “Deuteration” section captures the average deuteration value, the peptide’s average m/z value, and an opportunity to unselect the peptide if its peak characteristics are too poor for use. The “Deuteration – XIC Control” provides you with the opportunity to constrain how many isotopic peaks are used in the calculation. It also allows you to move around and resize the XIC peak integration window, which regenerates the mass spectrum. If the XIC peak picker has failed for some reason, you can input your own retention time as well. All XIC-related values are in minutes. When changes are made in properties, remember to select “Execute” at the bottom of the pane.

The main window, with the XIC viewer on the left and the mass spectrum viewer on the right, updates with all the parameter changes. A new XIC peak can be selected by clicking and dragging directly on the XIC graph display, which will again update the spectrum and the data in “Properties”. The actual isotopic peaks used in the calculation are indicated highlighted peak labels. The user can select or unselect isotopic peaks by clicking on the respective labels above the peaks. Both the XIC peaks and the isotopic distributions can be zoomed and un-zoomed.

The visualization and validation format is very useful in two situations. First, when initiating an experiment and building a peptide list from a standard output like Mascot search reports, it is worth the effort to modify the list through an assessment of each candidate peptide. Not every peptide from an MS/MS identification exercise will be of sufficient quality for conventional HX-MS analysis. Parsing the full list of peptides on a single test run can generate a revised peptide list that can be reliably used for subsequent analysis of a large set of additional HX-MS runs. Second, after a large set of runs has been processed, the Validation Viewer allows you to quickly check entries in the results table that appear unusual for some reason, such as smoothing or peak detection errors.

Processing MS/MS data for targeted HX-MS^2 analysis in the Studio

Project creation wizard

This tutorial is very similar in most parts to Tutorial 1 (HX-MS), therefore we will highlight the main differences. To start, select "HX-MSMS DDA" as the appropriate experiment type for LC-MS/MS data analysis, in which a survey MS scan is followed by any number of dependent MS/MS scans. During the project setup, the only difference is the "Select Mascot File" view which allows the user to define MS/MS specific settings (**Figure T9**). The first field allows the user to upload an optional .MGF file. This is used to define which fragments should be pre-selected for analysis for each peptide. Another important field in this view is the "Fragmentation Method" which must be set accordingly. Skipping the setting of the "Fragmentation Method" field will select "CID" with hydrogen scrambling as the default method.

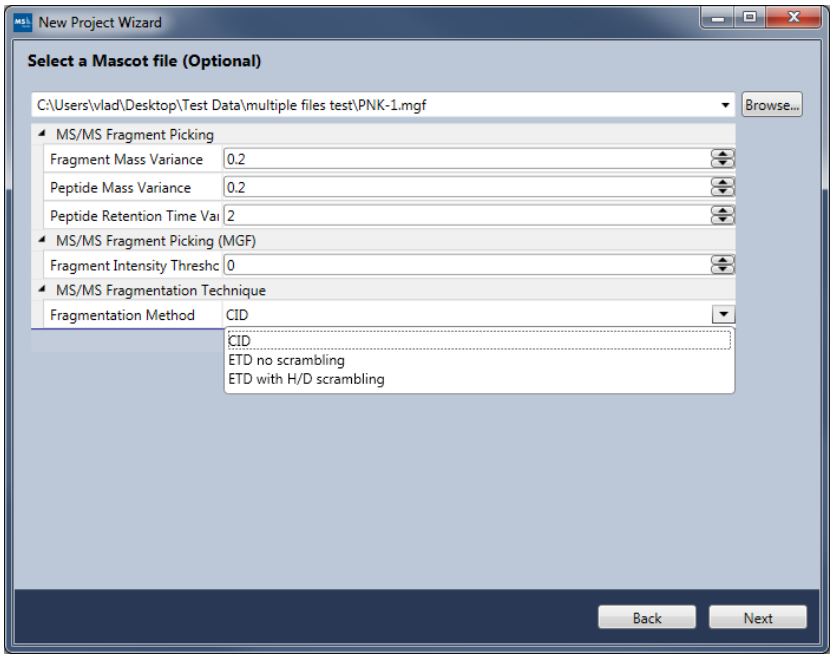


Figure T9. The "Select Mascot File" view.

Once the project is successfully configured, the main view will load, which is the same as for the "HX-MS" project type. If a mascot file was uploaded, you may review the fragments that were pre-selected by clicking on each peptide in the peptide list. You may modify the selected fragments before proceeding to the processing step (**Figure T10**).

Processing

In order to extract fragment deuteration, chose "Peptide HX-MS/MS" from the processing window. The "Peptide HX-MS" process is operable on the MS data and functions as if the data does not contain any fragment MS/MS as in a typical HX-MS project. Aside from the usual parameters of the "Peptide HX-MS", the MS/MS algorithm contains additional processing steps for extracting and analysis fragments. The most notable difference is the "MS/MS Spectrum Generator" step. The parameters for this step allow control over how MS/MS fragments are extracted, by defining the RT width, the m/z range, and

the class of fragments in which you are interested (e.g. sequence ions). All other fragments in the same MS/MS space are ignored.

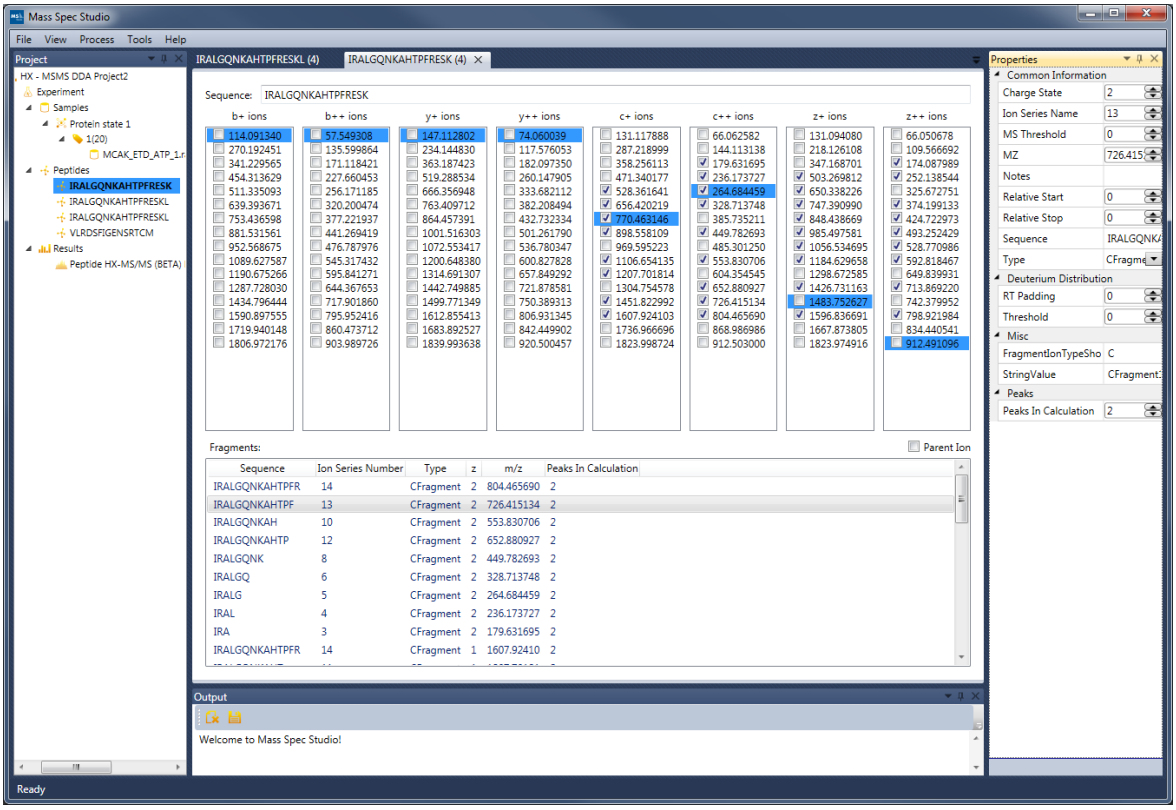


Figure T10. The peptide view where the user can manually select or de-select fragments for analysis. This view is activated upon clicking on a peptide under the “Peptides” tab.

Visualization and Validation

In addition to the usual validation of the peptide XIC and MS spectra, you have the option of validating each fragment of a given peptide in the same manner (**Figure T11**). It requires ensuring that the appropriate peaks are selected for computing deuteration values from a fragment. The “Reprocess” button triggered on each fragment will re-compute the deuteration after any unsaved changes. Once the data is processed and validated, it can be exported in CSV format from the File -> Export menu button. The resulting file will contain each peptide’s deuteration value as well as other information on a per replicate basis. The fragment data will be included in the resulting file with associations being made to the ID of the parent peptide via the “Parent ID” header.

Figure T11. Visualization of both MS and MS/MS data streams from DDA or product ion LC-MS/MS experiments. Chromatographic features are selected for generation of the peptide MS spectral signature, and the associated fragment plot is generated (top right) for the same region of retention time space, using criteria established in the project creation step. The zoomed-in mass spectrum of each fragment can be highlighted and peak characteristics interrogated. Changes can be executed to revise the results and captured in the validated peptide set.

Data

We included a test data set to demonstrate core functionality by way of the HX-MS package in the Mass Spec Studio. The data is a set of LC-MS runs of human PNK-1 protein under moderate deuteration conditions, acquired with an AB Sciex TripleTOF® 5600. The package contains 6 data files and 1 peptide CSV file. The data files can be split into two sets of replicates: 'free' and 'bound'. The 'free' set represents the PNK-1 protein in its un-bound state, whereas the 'bound' set represents interaction between PNK and the Ligase4 (587-844)+XRCC4 complex.

The data can be downloaded from the following link:

https://www.dropbox.com/s/nz0rlvawhxkiorr/Demo-PNK-Data.zip

References

1. Kessner, D., Chambers, M., Burke, R., Agus, D. & Mallick, P. ProteoWizard: open source software for rapid proteomics tools development. *Bioinformatics* **24**, 2534-2536 (2008).
2. Slysz, G.W. et al. Hydra: software for tailored processing of H/D exchange data from MS or tandem MS analyses. *BMC Bioinformatics* **10**, 162 (2009).
3. Gillet, L.C. et al. Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis. *Mol Cell Proteomics* **11**, O111 016717 (2012).
4. Craig, R., Cortens, J.P. & Beavis, R.C. Open source system for analyzing, validating, and storing protein identification data. *Journal of Proteome Research* **3**, 1234-1242 (2004).
5. Dominguez, C., Boelens, R. & Bonvin, A.M. HADDOCK: a protein-protein docking approach based on biochemical or biophysical information. *J Am Chem Soc* **125**, 1731-1737 (2003).