iSanXoT

User Guide:
Help and documentation

(1.2.7)

Table of contents

Introduction

iSanXoT is a standalone application for statistical analysis of mass spectrometry-based quantitative proteomics data. iSanXoT builds upon SanXoT [1], our previous publicly available implementation of the weighted spectrum, peptide, and protein (WSPP) statistical model [2] using the Generic Integration Algorithm (GIA) [3].

iSanXoT executes several kind of workflows for quantitative high-throughput proteomics, **systems biology** and the **statistical analysis**, **integration and comparison of experiments**.

iSanXoT was developed by the ***Cardiovascular Proteomics Lab***/***Proteomic Unit*** at **The National Centre for Cardiovascular Research** (CNIC, https://www.cnic.es).

Download

The multiple releases are available in the "release" section, located in the following link:

https://github.com/CNIC-Proteomics/iSanXoT/releases

Installation

Available operating systems

iSanXoT maintains the following operating systems and architectures and may add additional ones in the future:

Windows 10 Pro (x64)

MacOs High Sierra (10.13.6)

Ubuntu 20.04 (x64)

For more details, read the “Installation” section in the iSanXoT wiki:

https://github.com/CNIC-Proteomics/iSanXoT/wiki/Installation

Getting Started

This chapter describes iSanXoT’s graphical user interface and how to set up an analysis with iSanXoT.

For more details, read the “Getting Started” section in the iSanXoT wiki:

https://github.com/CNIC-Proteomics/iSanXoT/wiki/Getting_started

Modules

The iSanXoT desktop application houses a number of modules based on the SanXoT software package [1]. The information required to setup and execute every module is provided in a task-table.

There are four types of modules:

- Relation tables is a module that creates the relation tables used by the iSanXoT modules.
- The Basic modules call the individual scripts included in the SanXoT software package [1].
- The Compound modules perform a sequence of consecutive integrations based on the weighted spectrum, peptide and protein (WSPP) statistical model [2] and the systems-biology triangle (SBT) algorithm [3].
- Finally, there are two Reports: REPORT generates report files displaying the quantitative results produced by the above Basic and Composite modules when a workflow is executed; SANSON generates a similarity graph showing relationships between functional categories on the basis of the protein elements they share.

For more details, read the “Modules” section in the iSanXoT wiki:

https://github.com/CNIC-Proteomics/iSanXoT/wiki/Modules

Input Adaptor

The iSanXoT Input Adaptor offers users the possibility to either provide their own Identification/Quantification file, which contains the identification and quantification data, or have the Input Adaptor prepare this file from the results obtained using any of the mainstream proteomics pipelines.

For further details, read the “Input Adaptor” section in the iSanXoT wiki:

https://github.com/CNIC-Proteomics/iSanXoT/wiki/Input_adaptor

Sample Workflows

We describe in detail four sample workflows that illustrate the capacity of iSanXoT to statistically ascertain abundance changes in both multiplexed, isotopically labeled [3-5] and label-free [6] proteomics experiments.

- Workflow 1: One-step quantification in a labeled experiment ()
- Workflow 2: Step-by-step quantification and sample combination in a labeled experiment.
- Workflow 3: Quantification of posttranslationally modified peptides in a labeled experiment.
- Workflow 4: Label-free quantification.

For further details, read each section in the iSanXoT wiki.

More Documents

User Guide (v1.2.7):

https://github.com/CNIC-Proteomics/iSanXoT/wiki/docs/user_guides/User_Guide_iSanXoT-1.2.7.pdf

Presentations:

- WSPP demo:

  https://github.com/CNIC-Proteomics/iSanXoT/wiki/docs/presentations/WSPP_Demo_iSanXoT_1.0.X.pdf

- Input Adaptor demo:

  https://github.com/CNIC-Proteomics/iSanXoT/wiki/docs/presentations/Input_Adaptor_iSanXoT_1.0.X.pdf

License

This application is licensed under a **Creative Commons Attribution-NonCommercial-NoDerivs 4.0 Unported License**.

https://creativecommons.org/licenses/by-nc-nd/4.0/

You are free to:

Share - copy and redistribute the material in any medium or format.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms

https://github.com/CNIC-Proteomics/iSanXoT/wiki/License

Installation

Download

The multiple releases are available in the "release" section, located in the following link:

https://github.com/CNIC-Proteomics/iSanXoT/releases

Available operating systems

iSanXoT maintains the following operating systems and architectures and may add additional ones in the future:

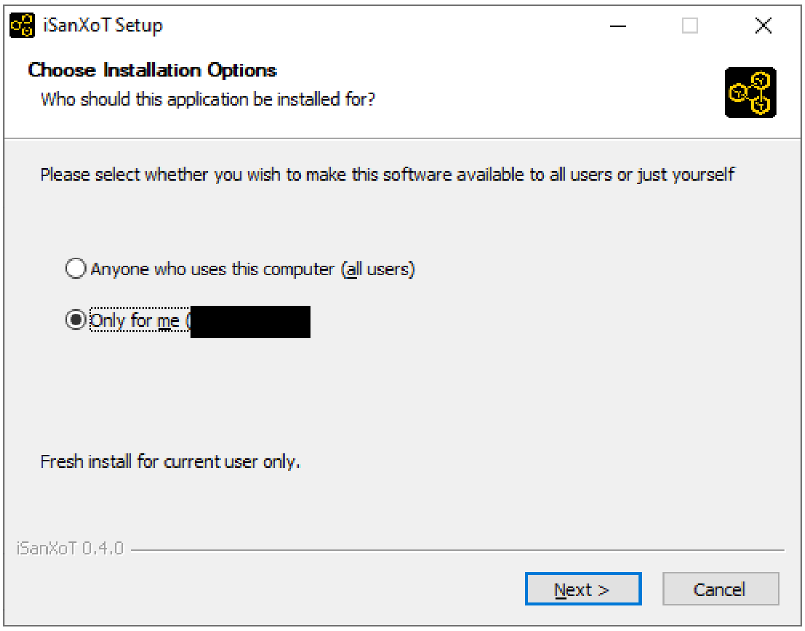
- Windows 10 Pro (x64)
- MacOs High Sierra (10.13.6)
- Ubuntu 20.04 (x64)

Windows distribution

The iSanXoT Windows distribution is packaged in a NSIS Launcher (exe file).

Download the exe Launcher: iSanXoT_Launcher_1.X.X.win32-x64.exe

Double-click the Launcher file; the Installer window will show up:



Choose Installation Options

Who should this application be installed for?

Please select whether you wish to make this software available to all users or just yourself

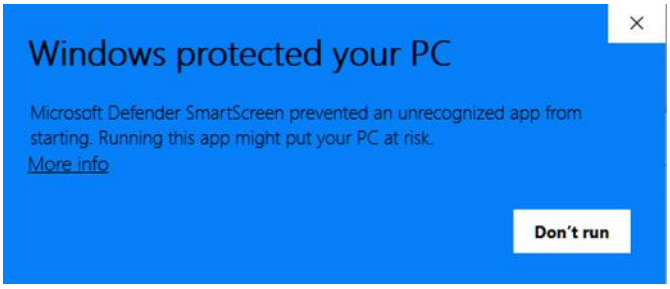
○ Anyone who uses this computer (all users)

● Only for me ([redacted]

Fresh install for current user only.

iSanXoT 0.4.0

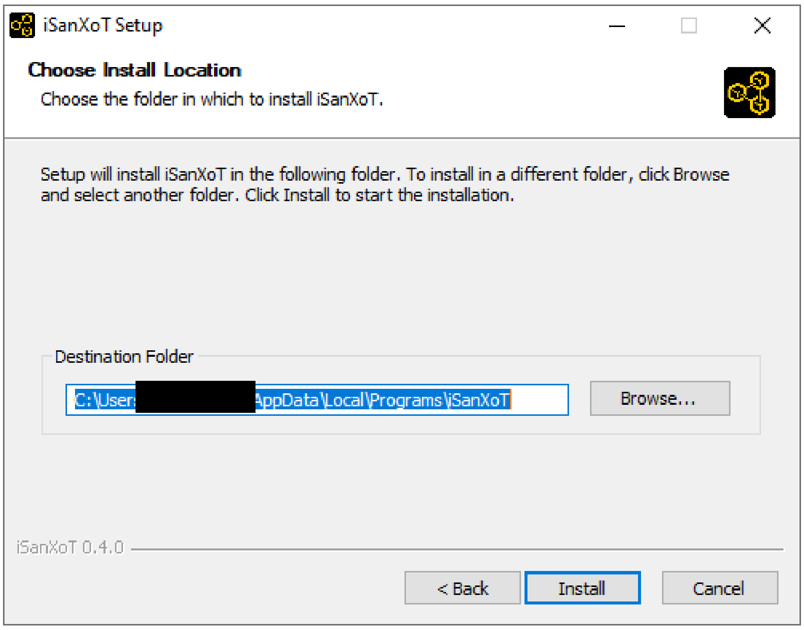
WARNING: Windows Defender SmartScreen might show a prompt suggesting that you cancel the installation; in such case click “More info” and then select the “Run anyway” option.



Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

Then, you can choose the iSanXoT installation folder:



iSanXoT Setup

Choose Install Location

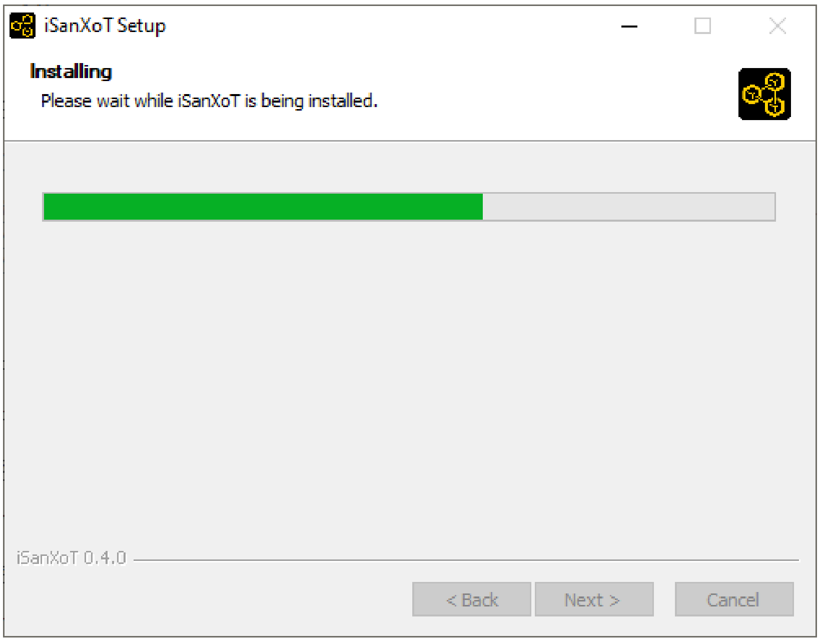
Choose the folder in which to install iSanXoT.

Setup will install iSanXoT in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Users[redacted]AppData\Local\Programs\iSanXoT

iSanXoT 0.4.0

Wait while iSanXoT is being installed.



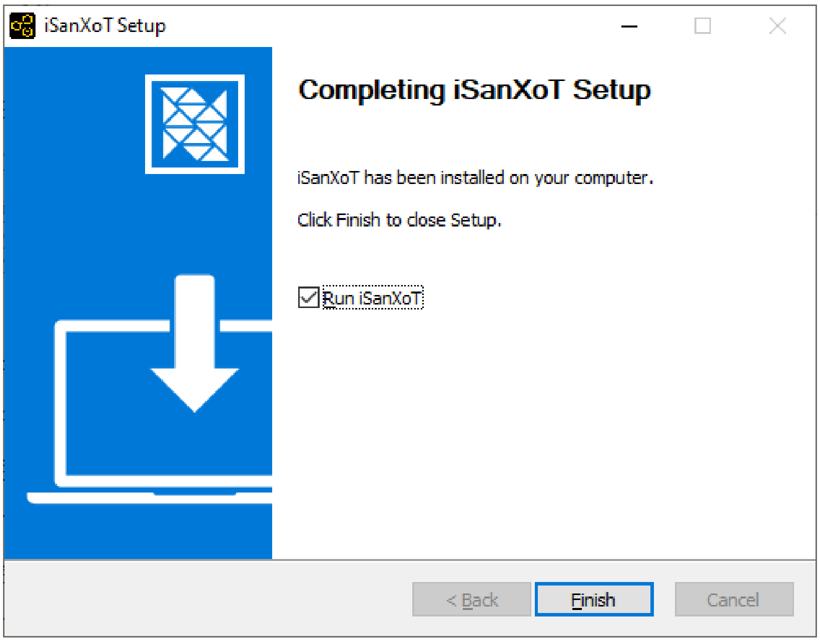
iSanXoT Setup

Installing

Please wait while iSanXoT is being installed.

iSanXoT 0.4.0

Once the installation has been completed, you are ready to run.



iSanXoT Setup

Completing iSanXoT Setup

iSanXoT has been installed on your computer.

Click Finish to close Setup.

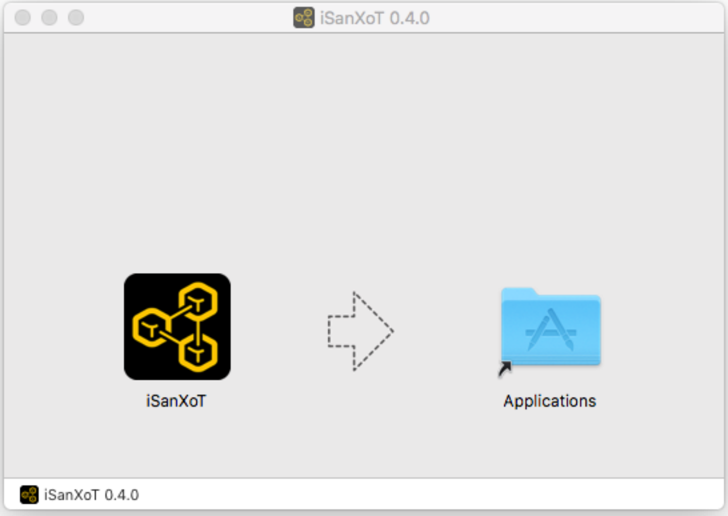
☑ Run iSanXoT

MacOS distribution

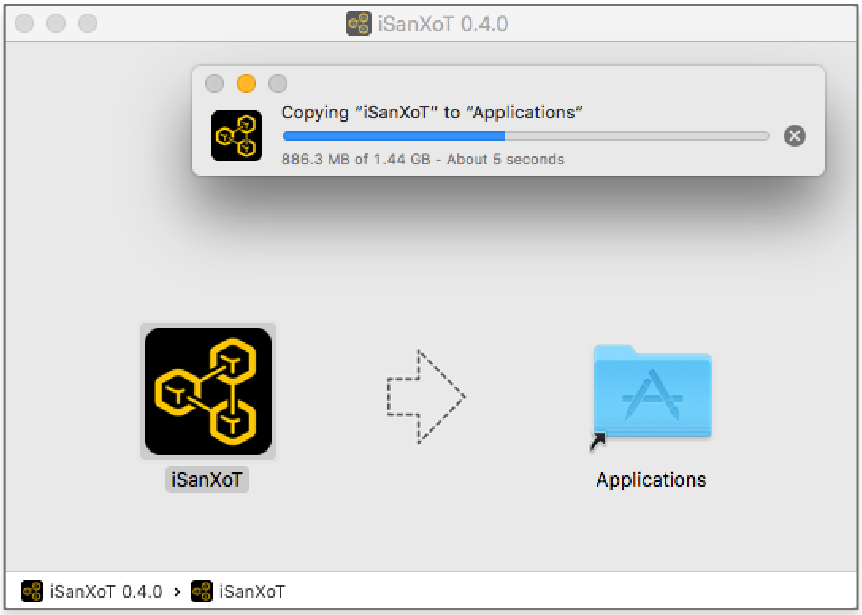
The iSanXoT MacOS distribution is packaged in a DMG container.

Download the DMG file: iSanXoT_Launcher_1.X.X.darwin-x64.dmg

Double-click the DMG file, then a Finder window will show up. This window will usually display iSanXoT's installer icon and a shortcut to the Applications folder, together with some sort of linking arrow:



Simply drag the iSanXoT icon to your Applications folder...



Copying "iSanXoT" to "Applications"

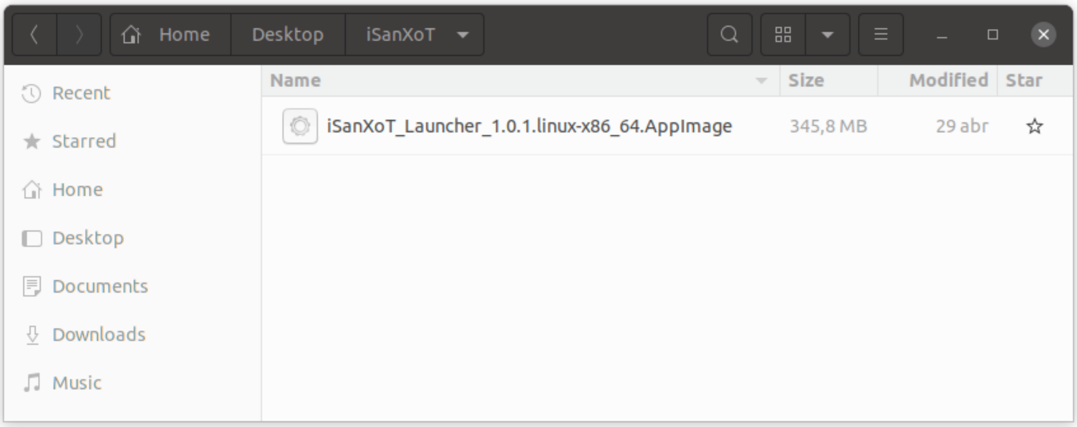
886.3 MB of 1.44 GB - About 5 seconds

and you're done: the iSanXoT application is now installed.

Linux distribution

The iSanXoT Linux distribution is packaged in an *AppImage*.

Download the AppImage: iSanXoT_Launcher_1.X.X.linux-x86_64.AppImage



Name	Size	Modified	Star
iSanXoT_Launcher_1.0.1.linux-x86_64.AppImage	345,8 MB	29 abr	☆

The AppImage file is just the application's compressed image. When executed, the application is mounted in a temporal folder. However, for the correct behaviour, the application has to be extracted to the "squashfs-root" folder in the current working directory using:

```
./iSanXoT_Launcher_1.X.X.linux-x86_64.AppImage --appimage-extract
```

Then, launch the iSanXoT application using:

```
squashfs-root/AppRun
```

Getting Started

This chapter describes iSanXoT's graphical user interface and how to set up an analysis with iSanXoT.

Opening the iSanXoT application

To open the iSanXoT application:

- In Windows: from the Start menu choose *Programs* > *iSanXoT*; or double-click the iSanXoT desktop icon.
- In MacOS: double-click the iSanXoT icon from the Applications folder.
- In Linux: from the *AppImage* file.
  The contents are extracted to the "squashfs-root" directory in the current working directory using:

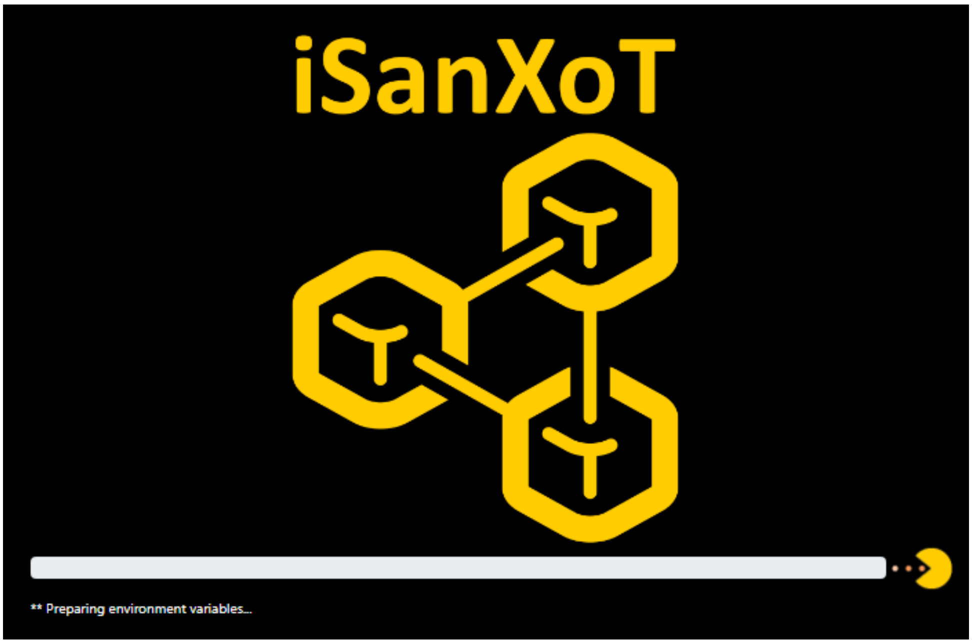
  ./iSanXoT_Launcher_1.x.linux-x86_64.AppImage --appimage-extract

  Now you can launch the iSanXoT application:

  squashfs-root/AppRun [...]

Installing required packages

The first time iSanXoT is run a window will show up displaying a progress bar to inform you about the percentage of packages that have been installed. These packages contain the libraries required by iSanXoT's backend, and are installed the first time you open the application.



** Preparing environment variables...

Figure 1. Installation window.

Closing the iSanXoT application

WARNING: If valid changes were made to your project, make sure to save it before quitting iSanXoT, as any changes will be lost otherwise (you won't be prompted for saving upon closing).

> To close the iSanXoT application:
>
> In Windows and Linux: choose *Project* > *Exit*, or click the X in the upper right corner of the main iSanXoT window.
>
> In Mac: choose *iSanXoT (menu)* > *Exit*, or click the red X in the upper left corner of the main iSanXoT window.

A dialog window will show up asking you to confirm the application closing. Click "Yes" if you really want to quit iSanXoT.

iSanXoT Projects

An iSanXoT project is primarily a container used to structure the data coming from your input file(s) and your workflow. The input file contains the identification and quantification data (for further details see the Input Adaptor Section). These fully-customisable workflows can perform quantitative proteomics analysis, systems biology analysis, and comparison and merging of experimental data from technical or biological replicates.

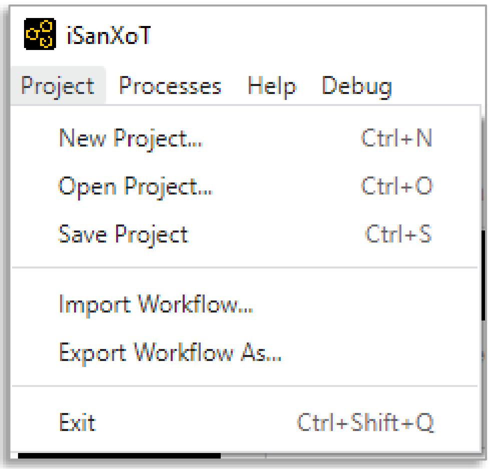
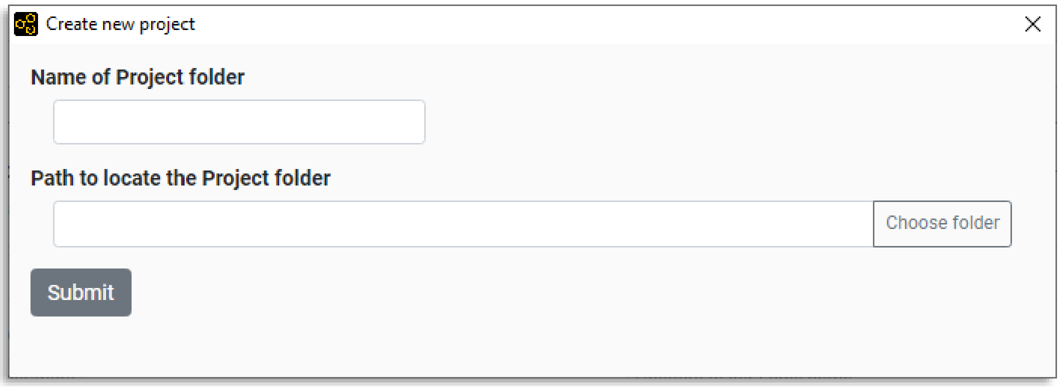


Figure 2. Project menu.

Creating a new project

Creating a project is the first step when conducting an analysis with iSanXoT. By selecting *Project > New Project* a window will show up where you can provide a name for the project as well as select a project folder where iSanXoT output files will be stored.


Create new project

Name of Project folder

Path to locate the Project folder

Figure 3. Window that creates a new project.

Opening a project

By selecting *Project > Open Project* as a folder selection dialog box shows up that allows the user to indicate the location of an already existing project folder to be opened by iSanXoT.

iSanXoT Main Window

The iSanXoT main window consists of an overhead menu, adaptor and module tabs, and content and execution panels (**Figure 4**).

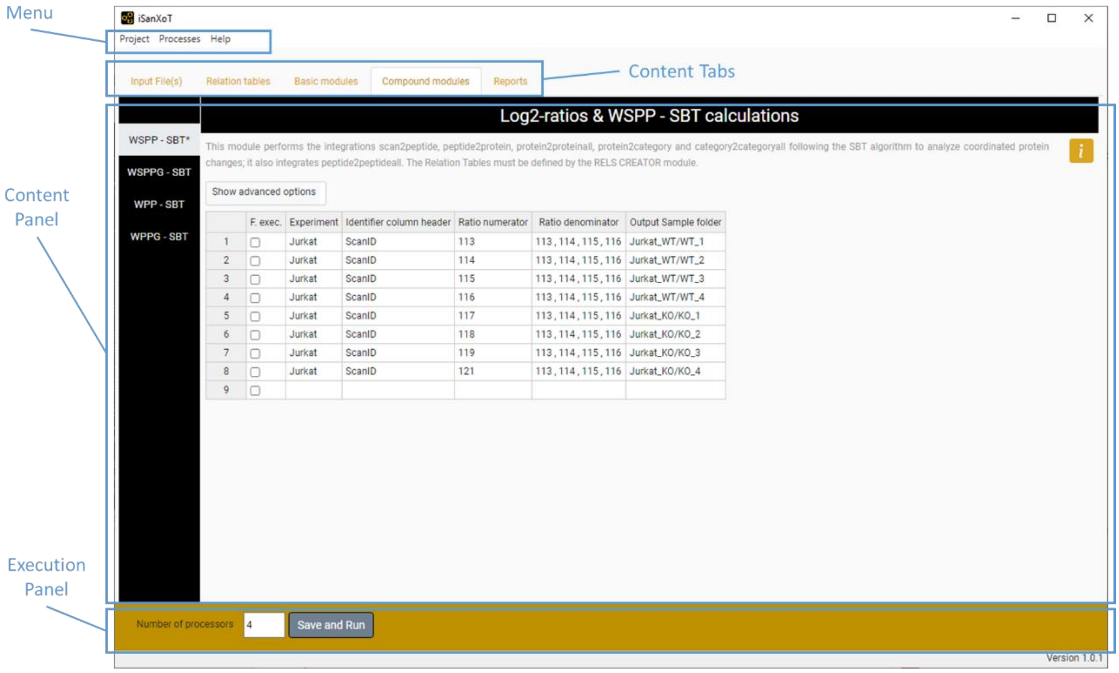
Menu

The Menu contains the following items:

- Project: Allows operations related to projects and workflows.
- Processes: Links to real-time display of the processes currently in execution by iSanXoT (see Running Processes Section).
- Help: Houses the different Sections of iSanXoT help.

Content tabs

1. Five tabs are displayed in iSanXoT’s project page. The Input tab displays the Project folder, where iSanXoT output files are stored, as well as the Identification file used in the project (see Input Adaptor Section). The remaining four tabs give access to iSanXoT modules: Relation Tables, Basic Modules, Compound Modules, and Report Modules.



Log2-ratios & WSPP - SBT calculations

This module performs the integrations scan2peptide, peptide2protein, protein2proteinall, protein2category and category2categoryall following the SBT algorithm to analyze coordinated protein changes; it also integrates peptide2peptideall. The Relation Tables must be defined by the RELS CREATOR module.

	F. exec.	Experiment	Identifier column header	Ratio numerator	Ratio denominator	Output Sample folder
1	□	Jurkat	ScanID	113	113 , 114 , 115 , 116	Jurkat_WT/WT_1
2	□	Jurkat	ScanID	114	113 , 114 , 115 , 116	Jurkat_WT/WT_2
3	□	Jurkat	ScanID	115	113 , 114 , 115 , 116	Jurkat_WT/WT_3
4	□	Jurkat	ScanID	116	113 , 114 , 115 , 116	Jurkat_WT/WT_4
5	□	Jurkat	ScanID	117	113 , 114 , 115 , 116	Jurkat_KO/KO_1
6	□	Jurkat	ScanID	118	113 , 114 , 115 , 116	Jurkat_KO/KO_2
7	□	Jurkat	ScanID	119	113 , 114 , 115 , 116	Jurkat_KO/KO_3
8	□	Jurkat	ScanID	121	113 , 114 , 115 , 116	Jurkat_KO/KO_4
9	□					

Figure 4. Main View of iSanXoT.

Content panel

This panel houses the elements of the Input and Modules tabs, which can be accessed through the sidebar menu showing on the left side of the panel. A title and a brief description of the Input/Module element is provided, as well as help icon linking to additional information on the specific Input element or Module selected.

Execution panel

The execution panel, located on the bottom of the main window, allows the user to indicate the number of processors to be used by iSanXoT, 4 by default. The Start button launches the execution of the workflow shaped by the Input elements and Modules.

Importing and Exporting Workflows

A project is shaped by a workflow that instructs iSanXoT how to process the data provided by the input file(s). While the whole project, including workflow and data, can be saved as indicated below (see the Saving a project section), there is a way to import and export just the workflow structure using iSanXoT menu (**Figure 2**).

Import Workflow

This option allows to import the task-tables of a workflow. For that, you have to provide the folder where the workflow is saved.

Export Workflow

The export workflow saves the task-tables of a workflow in the folder indicated by the user.

Executing a Project

Once your project contains all the necessary input data and workflow elements, you can execute the workflow by clicking "Save and Run" in the "Execution panel" after indicating the number of processors to be allocated for iSanXoT (see Execution panel above).

Bear in mind that every time you click the "Save and Run" button to execute a workflow, the project is first validated for consistency and saved. To save a project without executing it, you must use the appropriate menu item as explained in the next Section.

Saving a project

The *Project > Save Project* option saves your project, which contains the input data and the workflow elements, to the "Project folder". The project files are saved in the ".isanxot" folder. WARNING: Do not manipulate or delete the information stored in the ".isanxot" folder; you risk losing your project.

Whenever iSanXoT is prompted to save a project, the corresponding workflow is first validated for consistency and won't be saved when failed. Neither will the workflow execute if it has not been validated previously.

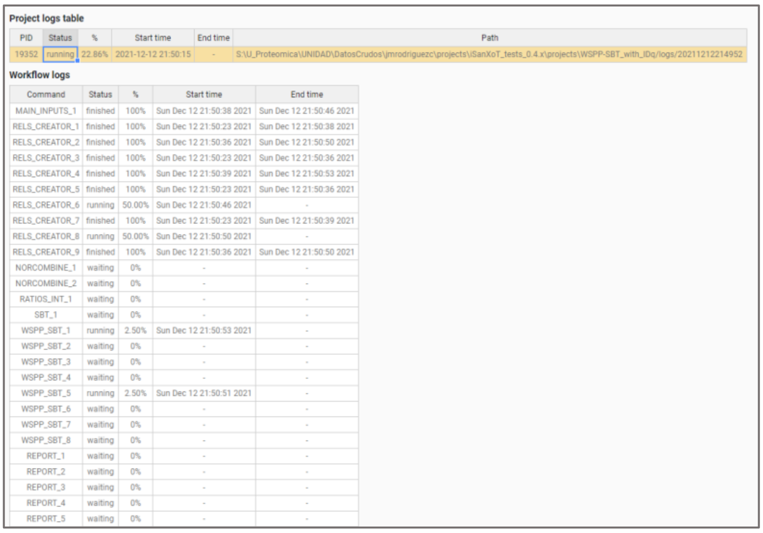
As well as the ".isanxot" folder, the following folders are necessary to shape your project:

- Exps, to store the files created by the Input Data adapters.
- Jobs, to store the *sample folders* of your workflow.
- Rels, to store the Relation Tables created by the RELS CREATOR module (see below).
- Reports, to store the Report files created by the REPORT module (see below).
- Stats, to store statistical data.
- Logs, to store the workflow execution log files.

Running Processes

When workflow execution successfully starts, a new window shows up displaying information about the processes currently running:

- The Project logs table shows project execution status. Several project executions can be monitored here, and the user must click a row to have the corresponding workflow logs displayed (see below).
- Workflow logs table displays status for the jobs set up in the workflow modules. If you click on a row you will see the trace log of the involved jobs (unless the job status is "cached").



Project logs table

PID	Status	%	Start time	End time	Path
19352	running	22.86%	2021-12-12 21:50:15	-	S:\U_Proteomica\UNIDAD\DatosCrudos\jmrodriguezc\projects\iSanXoT_tests_0.4.x\projects\WSPP-SBT_with_IDq/logs/20211212214952

Workflow logs

Command	Status	%	Start time	End time
MAIN_INPUTS_1	finished	100%	Sun Dec 12 21:50:38 2021	Sun Dec 12 21:50:46 2021
RELS_CREATOR_1	finished	100%	Sun Dec 12 21:50:23 2021	Sun Dec 12 21:50:38 2021
RELS_CREATOR_2	finished	100%	Sun Dec 12 21:50:36 2021	Sun Dec 12 21:50:50 2021
RELS_CREATOR_3	finished	100%	Sun Dec 12 21:50:23 2021	Sun Dec 12 21:50:36 2021
RELS_CREATOR_4	finished	100%	Sun Dec 12 21:50:39 2021	Sun Dec 12 21:50:53 2021
RELS_CREATOR_5	finished	100%	Sun Dec 12 21:50:23 2021	Sun Dec 12 21:50:36 2021
RELS_CREATOR_6	running	50.00%	Sun Dec 12 21:50:46 2021	-
RELS_CREATOR_7	finished	100%	Sun Dec 12 21:50:23 2021	Sun Dec 12 21:50:39 2021
RELS_CREATOR_8	running	50.00%	Sun Dec 12 21:50:50 2021	-
RELS_CREATOR_9	finished	100%	Sun Dec 12 21:50:36 2021	Sun Dec 12 21:50:50 2021
NORCOMBINE_1	waiting	0%	-	-
NORCOMBINE_2	waiting	0%	-	-
RATIOS_INT_1	waiting	0%	-	-
SBT_1	waiting	0%	-	-
WSPP_SBT_1	running	2.50%	Sun Dec 12 21:50:53 2021	-
WSPP_SBT_2	waiting	0%	-	-
WSPP_SBT_3	waiting	0%	-	-
WSPP_SBT_4	waiting	0%	-	-
WSPP_SBT_5	running	2.50%	Sun Dec 12 21:50:51 2021	-
WSPP_SBT_6	waiting	0%	-	-
WSPP_SBT_7	waiting	0%	-	-
WSPP_SBT_8	waiting	0%	-	-
REPORT_1	waiting	0%	-	-
REPORT_2	waiting	0%	-	-
REPORT_3	waiting	0%	-	-
REPORT_4	waiting	0%	-	-
REPORT_5	waiting	0%	-	-

Figure 5. View of running processes.

The running processes window can be also reached from the menu by selecting the “Processes > Main page” option.

Modules

The iSanXoT desktop application houses a number of modules based on the SanXoT software package [1]. The information required to setup and execute every module is provided in a task-table.

There are four types of modules:

- Relation tables is a module that creates the relation tables used by the iSanXoT modules.
- The Basic modules call the individual scripts included in the SanXoT software package [1].
- The Compound modules perform a sequence of consecutive integrations based on the weighted spectrum, peptide and protein (WSPP) statistical model [2] and the systems-biology triangle (SBT) algorithm [3].
- Finally, there are two Reports: REPORT generates report files displaying the quantitative results produced by the above Basic and Composite modules when a workflow is executed; SANSON generates a similarity graph showing relationships between functional categories on the basis of the protein elements they share.

Relation tables module

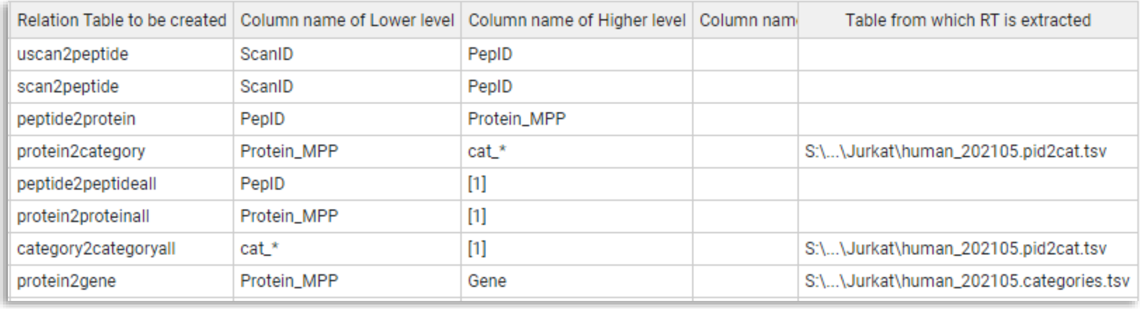
RELS CREATOR

This module generates relation tables (RT) from tab-separated values (TSV) files. Relation tables, which are TSV files relating lower level identifiers (e.g. peptides) to the corresponding higher level elements (e.g. proteins), are required for module execution. For this reason, the naming convention for the file indicated under “Relation Table to be created” in the RELS CREATOR task table is *lower level + “2” + higher level*; e.g. whether an integration is from the “peptide” level to the “protein” level, a relation table called “peptide2protein” will be necessary.

The fields showing in the RELS CREATOR task table are (**Figure 6**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Relation Table to be created* specifies relation table filenames. As commented above, the naming convention for these files is *lower level + “2” + higher level*; e.g. whether an integration is from the “peptide” level to the “protein” level, a relation table called “peptide2protein” will be necessary.
- *Column name of Lower level* is the column header that designates which elements from the indicated file (see below) will be taken as lower level elements in the resulting relation table.
- *Column name of Higher level* is the column header that designates which elements from the indicated file (see below) will be taken as higher level elements in the resulting relation table.
- *Column name of 3rd column* is the column header that designates which elements from the indicated file (see below) will be taken as third column elements in the resulting relation table.

- Table from which RT is extracted is the full path name for the TSV file to be used to build the relation tables. If the cell is empty, the Input file (ID-q.tsv) is applied for iSanXoT workflow.



Relation Table to be created	Column name of Lower level	Column name of Higher level	Column nam	Table from which RT is extracted
uscan2peptide	ScanID	PepID		
scan2peptide	ScanID	PepID		
peptide2protein	PepID	Protein_MPP		
protein2category	Protein_MPP	cat_*		S:\...\Jurkat\human_202105.pid2cat.tsv
peptide2peptideall	PepID	[1]		
protein2proteinall	Protein_MPP	[1]		
category2categoryall	cat_*	[1]		S:\...\Jurkat\human_202105.pid2cat.tsv
protein2gene	Protein_MPP	Gene		S:\...\Jurkat\human_202105.categories.tsv

Figure 6. A sample Task-Table in the example of RELS CREATOR module.

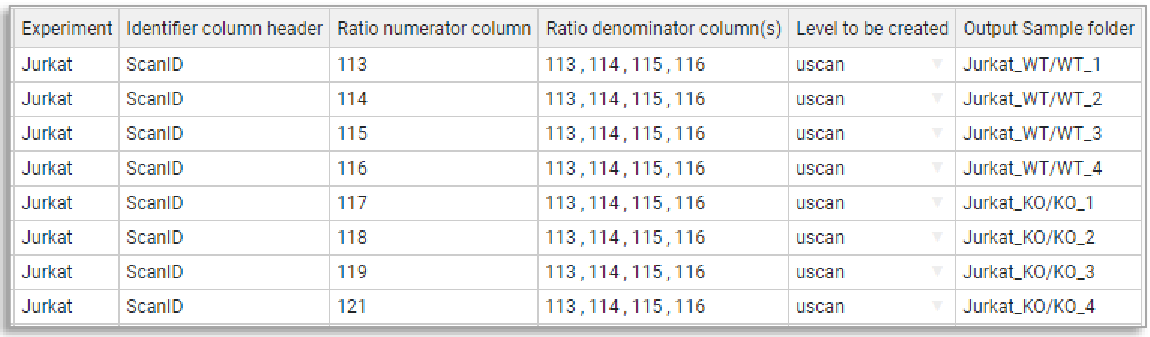
Basic modules

LEVEL CREATOR

This module creates levels, which are TSV files containing identifiers, log$_2$-ratio values and statistical weight values. The data are extracted from the Identification/Quantification file (see Input Adaptor section).

The following fields are displayed in the LEVEL CREATOR task table (**Figure 7**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Experiment* is the column header that designates which elements from the Identification file will be used to create the level indicated.
- *Identifier column header* is the identification file column header that unambiguously identifies the scans.
- *Ratio numerator column* specifies which column header from the identification file designates the quantitative values to be used as a numerator for the log$_2$-ratio calculation.
- *Ratio denominator column(s)* specifies which column header from the identification file designates the quantitative values to be used as a denominator for the log$_2$-ratio calculation.
- *Level to be created* designates the level name.
- *Output Sample folder* indicates the name of the folder where the level data file will be saved.



Experiment	Identifier column header	Ratio numerator column	Ratio denominator column(s)	Level to be created	Output Sample folder
Jurkat	ScanID	113	113 , 114 , 115 , 116	uscan	Jurkat_WT/WT_1
Jurkat	ScanID	114	113 , 114 , 115 , 116	uscan	Jurkat_WT/WT_2
Jurkat	ScanID	115	113 , 114 , 115 , 116	uscan	Jurkat_WT/WT_3
Jurkat	ScanID	116	113 , 114 , 115 , 116	uscan	Jurkat_WT/WT_4
Jurkat	ScanID	117	113 , 114 , 115 , 116	uscan	Jurkat_KO/KO_1
Jurkat	ScanID	118	113 , 114 , 115 , 116	uscan	Jurkat_KO/KO_2
Jurkat	ScanID	119	113 , 114 , 115 , 116	uscan	Jurkat_KO/KO_3
Jurkat	ScanID	121	113 , 114 , 115 , 116	uscan	Jurkat_KO/KO_4

Figure 7. A sample task-table in the LEVEL CREATOR module.

LEVEL CALIBRATOR

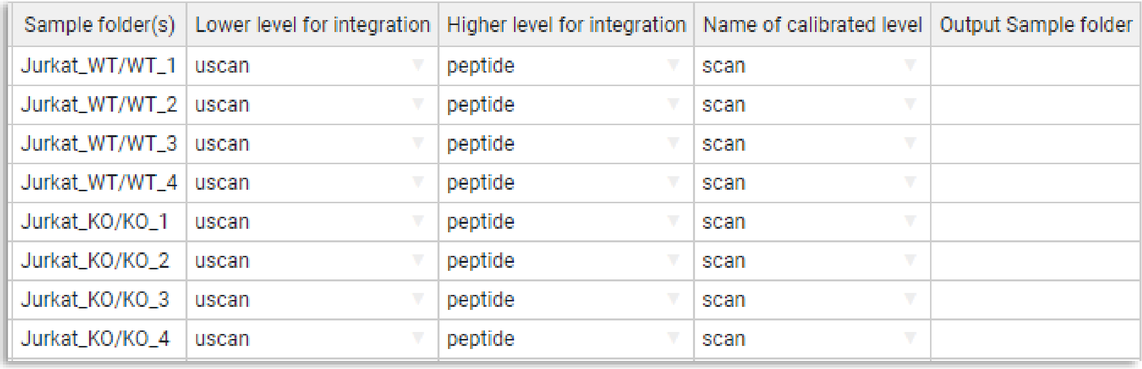
This module calibrates the above-described levels using the "Klibrate" program included in the SanXoT software package [1]. To perform the calibration, two parameters (weight constant and variance) are iteratively calculated using the Levenberg-Marquardt algorithm (for more details see the information about "Klibrate" in the SanXoT software package [1]).

The output calibrated level contains new statistic weight values for the identifier and $\log_2$-ratio elements displayed in the uncalibrated level data file. This is necessary for the levels to be used as inputs to the INTEGRATE module.

Standard parameters

The fields to be completed in this module are (**Figure 8**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Sample folder(s)* indicates the name(s) of the folder(s) containing the uncalibrated data file(s) that were previously generated by the LEVEL CREATOR module.
- *Lower level for integration* indicates which lower level elements are to be used in the integration carried out for the calibration.
- *Higher level for integration* indicates which higher level elements are to be used in the integration carried out for the calibration.
- *Name of calibrated level* is the name for the output data file containing the new, calibrated statistical weight values.
- *Output Sample folder* specifies the name of the folder where the output data file containing the new, calibrated statistical weight values will be saved. If the cell is empty, the output sample folder is the given "Sample folder" (second column).



Sample folder(s)	Lower level for integration	Higher level for integration	Name of calibrated level	Output Sample folder
Jurkat_WT/WT_1	uscan	peptide	scan	
Jurkat_WT/WT_2	uscan	peptide	scan	
Jurkat_WT/WT_3	uscan	peptide	scan	
Jurkat_WT/WT_4	uscan	peptide	scan	
Jurkat_KO/KO_1	uscan	peptide	scan	
Jurkat_KO/KO_2	uscan	peptide	scan	
Jurkat_KO/KO_3	uscan	peptide	scan	
Jurkat_KO/KO_4	uscan	peptide	scan	

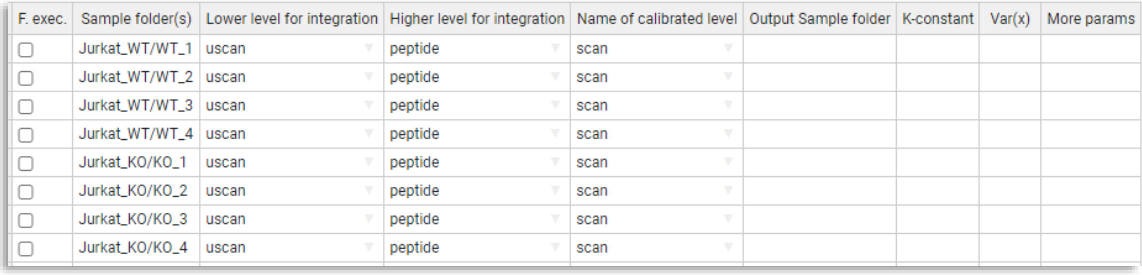
Figure 8. A sample task table in the LEVEL CALIBRATOR module.

Advanced parameters

The LEVEL_CALIBRATOR module accepts the following additional parameters (**Figure 9**):

To perform the calibration two parameters, have to be calculated: the k (weight constant), and the variance.

- *K-constant* sets a forced value for the k-constant. Using this parameter, the introduced value is forced as K-constant.
- *Var(x)* sets a forced value for the variance. Using this parameter, the introduced value is forced as the variance.
- *More params* allows adding more parameters to the internal programs of the module. For more details see More params in the "Special Parameters" Section.



F. exec.	Sample folder(s)	Lower level for integration	Higher level for integration	Name of calibrated level	Output Sample folder	K-constant	Var(x)	More params
☐	Jurkat_WT/WT_1	uscan	peptide	scan				
☐	Jurkat_WT/WT_2	uscan	peptide	scan				
☐	Jurkat_WT/WT_3	uscan	peptide	scan				
☐	Jurkat_WT/WT_4	uscan	peptide	scan				
☐	Jurkat_KO/KO_1	uscan	peptide	scan				
☐	Jurkat_KO/KO_2	uscan	peptide	scan				
☐	Jurkat_KO/KO_3	uscan	peptide	scan				
☐	Jurkat_KO/KO_4	uscan	peptide	scan				

Figure 9. A task-table displaying advanced parameters for the LEVEL CALIBRATOR module.

INTEGRATE

The INTEGRATE module performs statistical calculations based on the WSPP model by iteratively applying the generic integration algorithm (GIA) [3] on calibrated data files (**Figure 10**).

Integrations are carried out from lower level data to higher level data (e.g. from the peptide level to the protein level and from the protein level to the gene level).

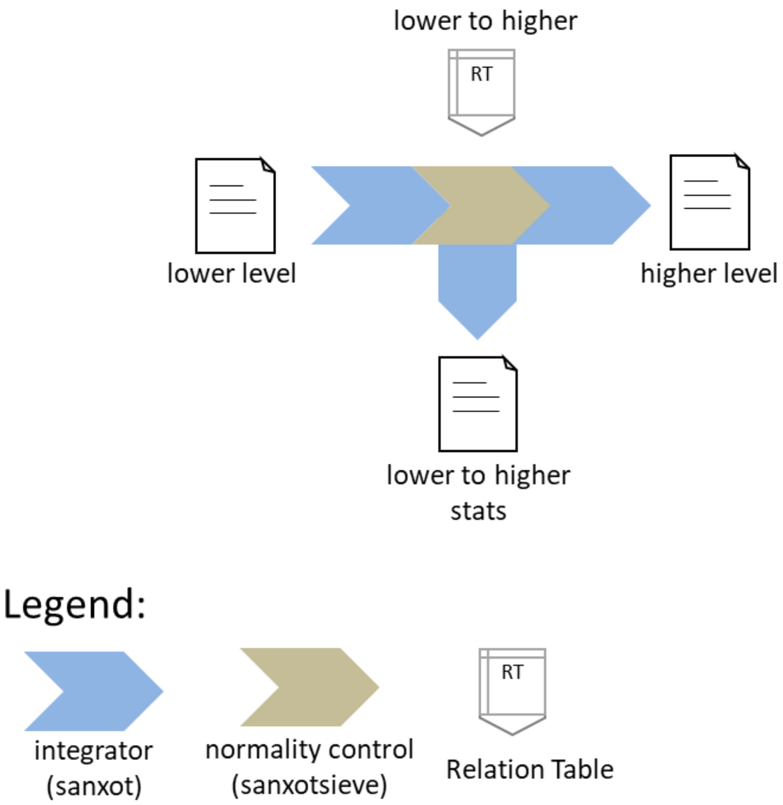


Figure 10. Schematic representation of the INTEGRATE module. *The integration is carried out from any lower level to any higher level using the programs "SanXoT" and "SanXoTSieve" and the generic integration algorithm (GIA).*

More in detail, the INTEGRATE module needs two TSV files as inputs:

1. A data file containing three data columns: identifier (a text string that is used to unambiguously identify the low level elements), quantitative value (log_2-ratio of the two measurements to be compared) and statistical weight (a parameter that measures the accuracy of the quantitative value).
2. A relation table, which links the lower level identifiers to those in the higher level. This file contains two columns: higher level identifiers on the left and lower level identifiers on the right.

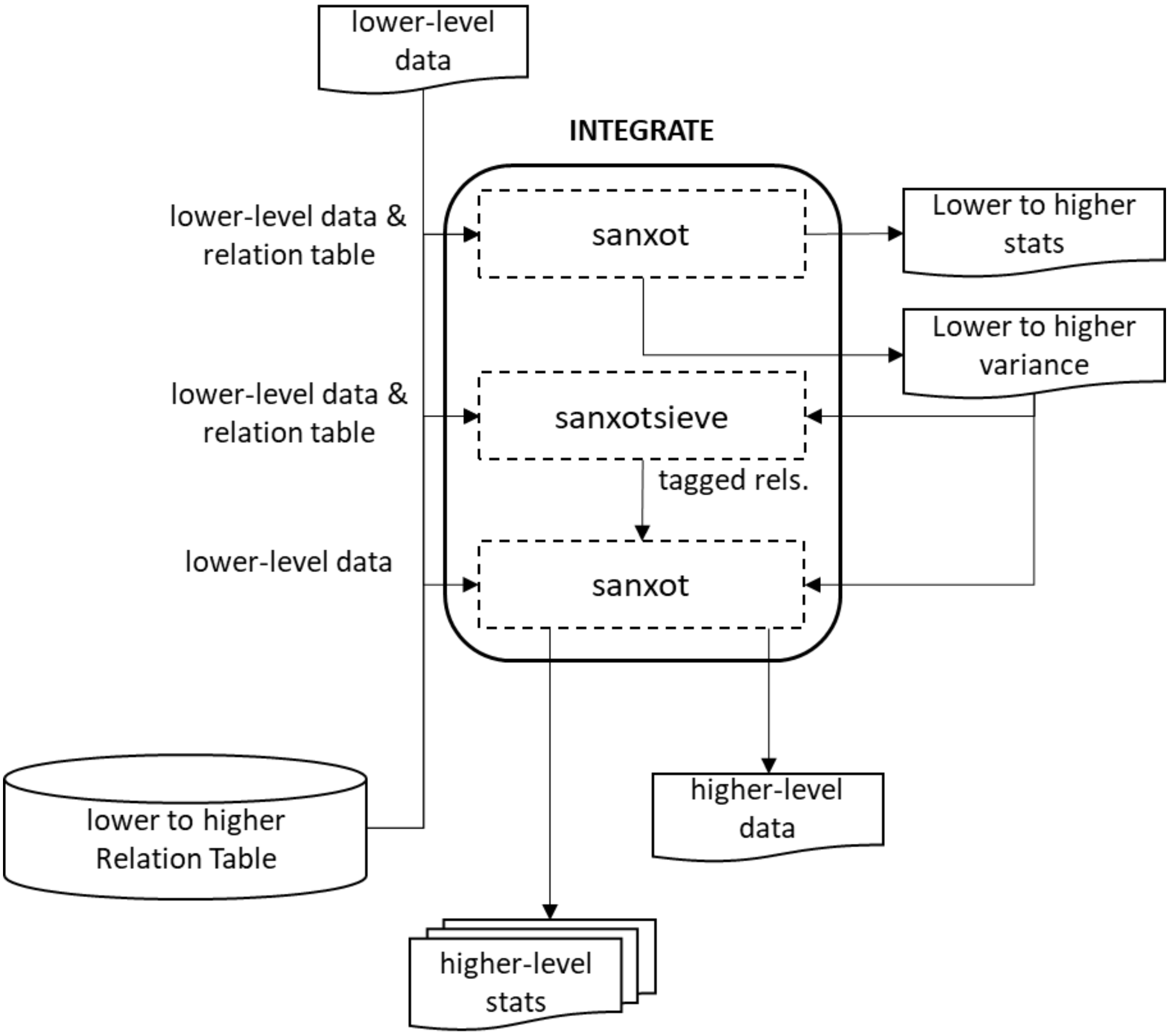


Figure 11. The INTEGRATE module flowchart. *A first integration is done with "SanXoT" that calculates the variance; then "SanXoTSieve" removes outliers tagging them in a new relation table; finally, a second integration is done with "SanXoT" using the variance calculated.*

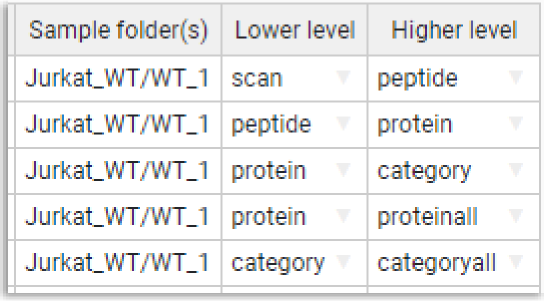
For every integration, the SanXoT program calculates the general variance using a robust iterative method. Then SanXoTSieve is used to tag outlier elements [2] by assessing the probability that a lower level element be a significant outlier of the standardized (i.e. N(0,1)) $\log_2$-ratio distribution. The most extreme outliers are thus removed sequentially and the integration repeated until all outliers below a user-defined false discovery rate (FDR) threshold have been removed. Finally, a second integration is carried out by SanXoT using the variance calculated in the first integration and discarding the outliers tagged in the new relation table (**Figure 11**).

The output data files generated by INTEGRATE contain the quantitative data for the higher level and can be used as inputs to other modules; in addition, each integration generates several additional files which contain information about the integration. For further details see SanXoT documentation [1].

Standard parameters

The parameters to be provided in the INTEGRATE module task table are (**Figure 12**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Sample folder(s)* indicates the names of the folder(s) where the lower level data file is located.
- *Lower level* indicates the name of the lower level data file to be used. This file contains three data columns: identifier, quantitative value and statistical weight.
- *Higher level* indicates the name of the higher level to which the lower level elements will be integrated.



Sample folder(s)	Lower level	Higher level
Jurkat_WT/WT_1	scan	peptide
Jurkat_WT/WT_1	peptide	protein
Jurkat_WT/WT_1	protein	category
Jurkat_WT/WT_1	protein	proteinall
Jurkat_WT/WT_1	category	categoryall

Figure 12. A sample task-table in the INTEGRATE module.

Advanced parameters

The INTEGRATE module accepts the following additional parameters (**Figure 13**):

- *Output Sample folder* is the name of the folder where the level data and statistics are saved.
- *Tag* is a text label that indicates which elements from the lower level are integrated into the higher level. The tags must be specified in the third column of the corresponding Relation Table. This allows the user to discard elements for integration without needing to eliminate them from the Relation Table. Thus, if the label “marked” is used as a Tag, only the lower level elements containing the label marked in the third column of the lower_level2higher_level Relation Table will be integrated. Logical operators can also be used in the Tag field to make complex decisions.

  Tag is a parameter to distinguish groups to perform the integration. For instance, if the user specifies “marked” the elements containing the label “marked” in the third column of the “lower_level2higher_level” Relation Table will be included in (or discarded from) the integration.
  The tag can be used by inclusion, such as "mod" or by exclusion, putting first the "!" symbol, such as "!mod". Tags should be included in a third column of the relations file.

  Different tags can be combined using logical operators "and" (&), "or" (|), and "not" (!), and parentheses. Some examples:

  !out&mod
  !out&(dig0|dig1)
  (!dig0&!dig1)|mod1

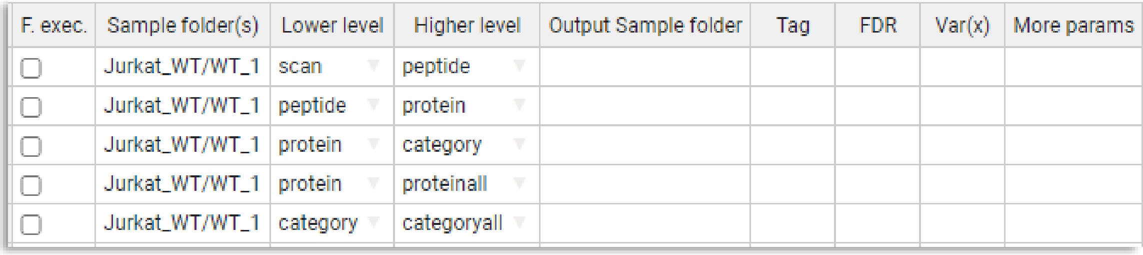
mod1|mod2|mod3

Warning: Unless specified otherwise by the user, by default iSanXoT eliminates outliers from the lower level according to an FDR<1% threshold.
iSanXoT automatically adds the tag "out" in the third column of the relation table to label outliers, so that they are not integrated. It is not thus recommended to use this tag for other purposes.

Note that although the discarded elements will not be included in calculations, the parameter Z will be calculated and tabulated in the corresponding output (outStats) file.

For further details see SanXoT wiki
(https://www.cnic.es/wiki/proteomica/index.php/SanXoT_software_package).

- *FDR* is an FDR threshold other than the default value (0.01, i.e. 1%) for outlier removal. If "0" is specified as the FDR value, then no outliers will be discarded.
- *Var(x)* sets a fixed value for the variance. The default value (blank) means that the variance will be iteratively calculated based on the Levenberg-Marquardt algorithm in the first *Lower level-to-Higher level* integration.
- *More params* allows adding more parameters to the internal programs of the module. For more details see More params in the "Special Parameters" Section.



F. exec.	Sample folder(s)	Lower level	Higher level	Output Sample folder	Tag	FDR	Var(x)	More params
□	Jurkat_WT/WT_1	scan	peptide					
□	Jurkat_WT/WT_1	peptide	protein					
□	Jurkat_WT/WT_1	protein	category					
□	Jurkat_WT/WT_1	protein	proteinall					
□	Jurkat_WT/WT_1	category	categoryall					

Figure 13. A sample task-table displaying advanced parameters for the INTEGRATE module.

NORCOMBINE

The NORCOMBINE module combines technical or biological replicates (**Figure 14**). For example, NORCOMBINE can be used to merge the protein level data from 4 individual patients and 4 individual controls into a patient- and a control level protein dataset, respectively.

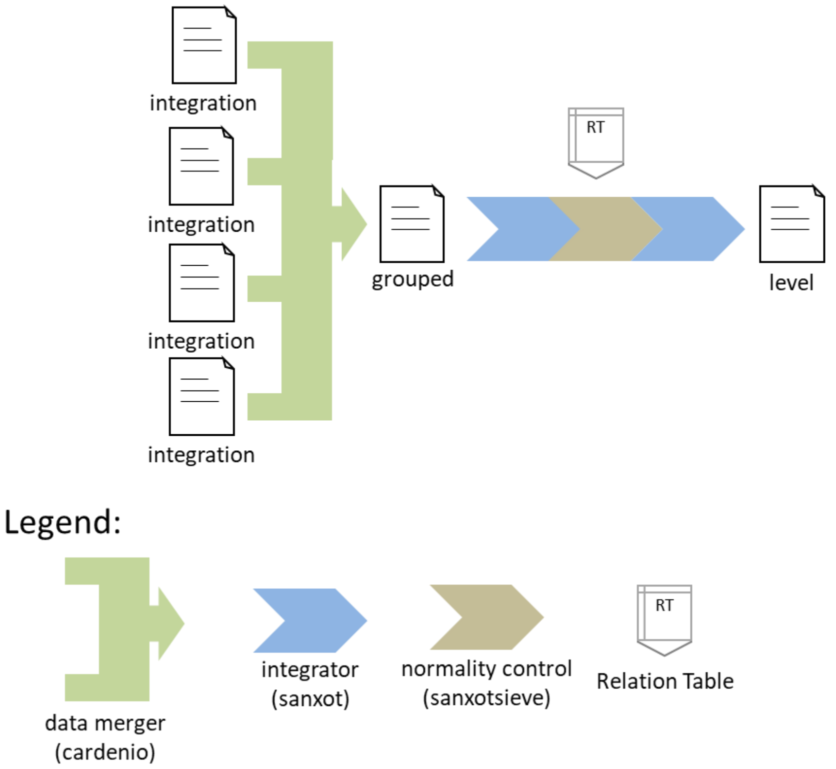


Figure 14. Schematic representation of the NORCOMBINE module used to combine technical or biological replicates.

Experiment merging relies on the “Cardenio” program from the SanXoT software package [1], which is used to generate merged data files and relation tables that are later integrated to the grouped level using “SanXoT” and “SanXoTSieve” (**Figure 14**).

NORCOMBINE requires the user to specify which *lowerNorm* files contain the necessary data for the samples to be combined. These *lowerNorm* files, previously generated by the INTEGRATE module, display the lower level identifiers on the left, followed by the corresponding centred $\log_2$-ratio values (i.e. the values obtained after subtracting the high level value) in the second column, and either the integration statistical weight (in the case of *lowerNormV*) or the variance (for *lowerNormW).* The SanXoT program “Cardenio” [1] is then used to generate merged data files and relation tables that are later integrated to the grouped level using “SanXoT” and “SanXoTSieve” (**Figure 15**).

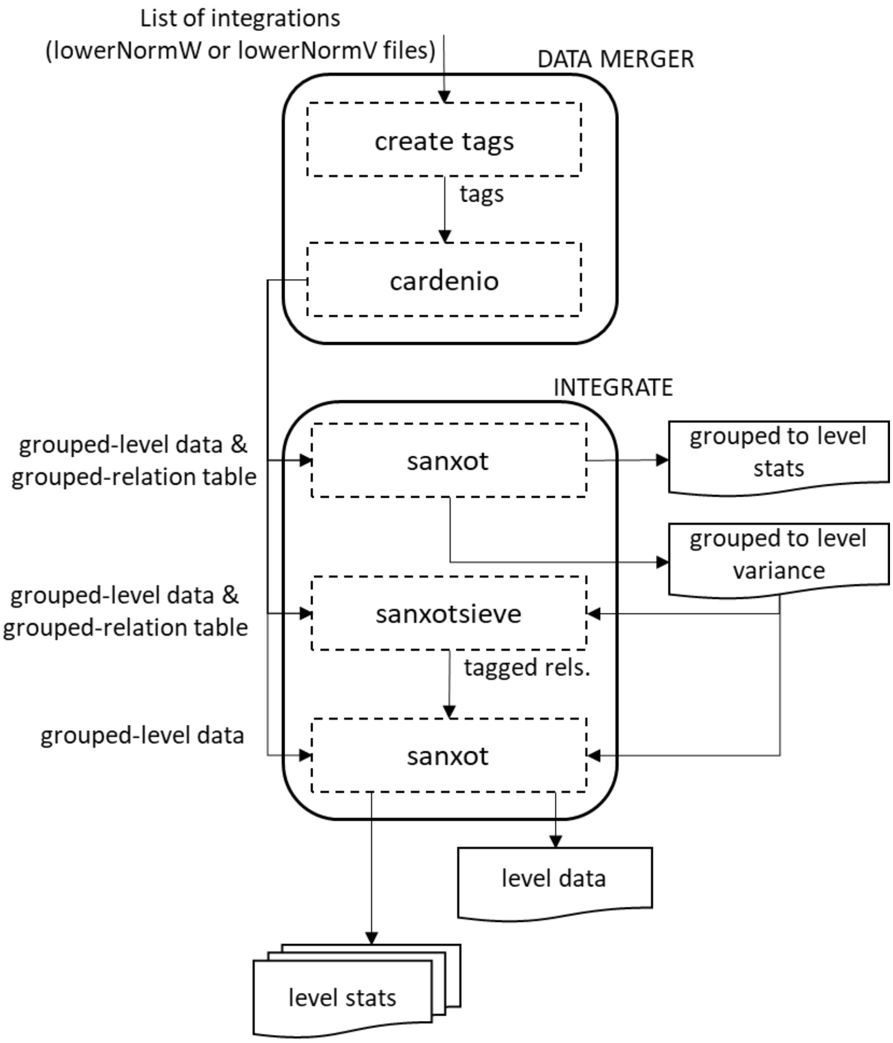
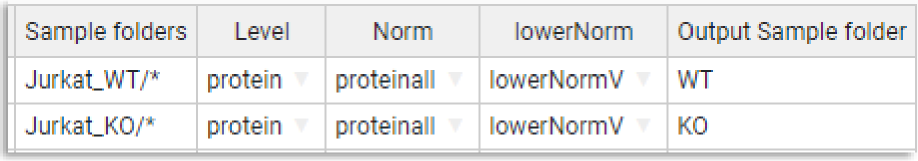


Figure 15. The NORCOMBINE module flowchart.

Standard parameters

The default NORCOMBINE module task table shows the following fields (**Figure 16**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Sample folders* indicates the names of the folder(s) containing the lower level data (samples) to be combined.
- *Level* indicates the type of elements to be combined (e.g. peptides or proteins).
- *Norm* specifies the normalization scheme to be used in the integrations.
- *lowerNorm* specifies the type of *lowerNorm* file (see above) to be used.
- *Output Sample folder* is the name of the folder where the grouped level data and statistics are saved.



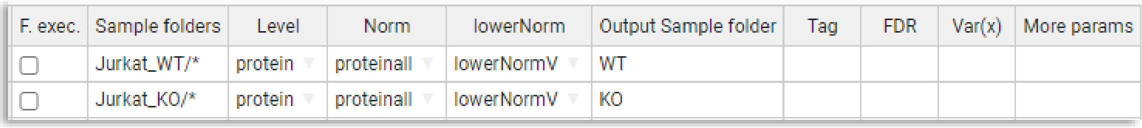
Sample folders	Level	Norm	lowerNorm	Output Sample folder
Jurkat_WT/*	protein	proteinall	lowerNormV	WT
Jurkat_KO/*	protein	proteinall	lowerNormV	KO

***Figure 16. A sample task-table in the NORCOMBINE module.** In this case, the asterisk wildcard has been used to select multiple sample folders.*

Advanced parameters

The NORCOMBINE module accepts the following additional parameters (**Figure 17**):

- *Tag* is a text label that indicates which elements from the lower level are integrated into the higher level. The tags must be specified in the third column of the corresponding Relation Table. This allows the user to discard elements for integration without needing to eliminate them from the Relation Table. Thus, if the label "marked" is used as a Tag, only the lower level elements containing the label marked in the third column of the lower_level2higher_level Relation table will be integrated. Logical operators can also be used in the Tag field to make complex decisions.
  By default, iSanXoT eliminates outliers from the lower level according to an FDR<1% threshold.
  For further details, read the Advanced Parameters for the INTEGRATE module.
- *FDR* is an FDR threshold other than the default value (0.01, i.e. 1%) for outlier removal. If "0" is specified as the FDR value, then no outliers will be discarded.
- Var(x) sets a fixed value for the variance. The default value (blank) means that the variance will be iteratively calculated based on the Levenberg-Marquardt algorithm.
- *More params* allows adding more parameters to the internal programs of the module. For more details see More params in the "Special Parameters" Section.



F. exec.	Sample folders	Level	Norm	lowerNorm	Output Sample folder	Tag	FDR	Var(x)	More params
☐	Jurkat_WT/*	protein	proteinall	lowerNormV	WT				
☐	Jurkat_KO/*	protein	proteinall	lowerNormV	KO				

Figure 17. A sample task table with advanced parameters in the NORCOMBINE module.

RATIOS

This module prepares the data file and relation table required as a first step in the calculation of a ratio defined by the user (e.g KO vs WT). For that, the new $\log_2$-ratio is calculated as the difference between numerator and denominator values, whereas the corresponding statistical weight is assessed according to the method indicated by the user in the V Method filed of the RATIOS task table (**Figure 18** and **Figure 19**):

- *max* uses the maximum value between the numerator and denominator statistical weight value.
- *form* uses the value resulting from 1/(1/Vn + 1/Vd), where Vn and Vd are the statistical weight value for the numerator and the denominator, respectively.

- *avg* uses the average value between the numerator and denominator statistical weight value.

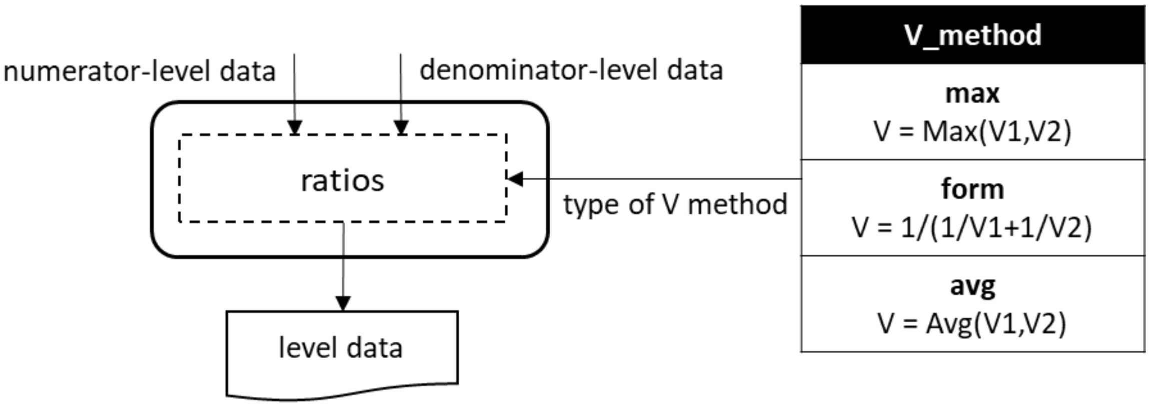
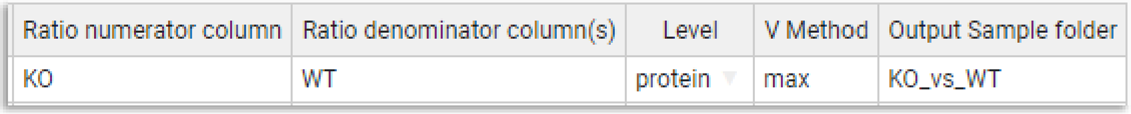


Figure 18. The RATIOS module flowchart.

The RATIOS module task table displays the following parameter fields (**Figure 19**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Ratio numerator column* specifies the name of the folder containing the quantitative value to be used as a numerator for the new log_2-ratio calculation.
- *Ratio denominator column(s)* specifies the name of the folder(s) containing the quantitative values to be used as a denominator for the new log_2-ratio calculation. The sample folders have to be separated by comma.
- *Level* designates the level (i.e. peptide, protein, gene or category) at which the ratio is to be calculated.
- *Output Sample folder* indicates the name of the folder where the resulting log_2-ratio and statistical weight values will be saved (e.g. KO_vs_WT).



Ratio numerator column	Ratio denominator column(s)	Level	V Method	Output Sample folder
KO	WT	protein	max	KO_vs_WT

Figure 19. A sample task-table in the RATIOS module.

SBT

This module is based on the Systems Biology Triangle (SBT) algorithm [3], which performs an integration between the lower and the higher levels using the variance previously obtained in an integration between the lower and an intermediate level (**Figure 20**). Usually the SBT module is applied to carry out the protein-to-grand mean integration using the variance associated with the protein-to-category integration.

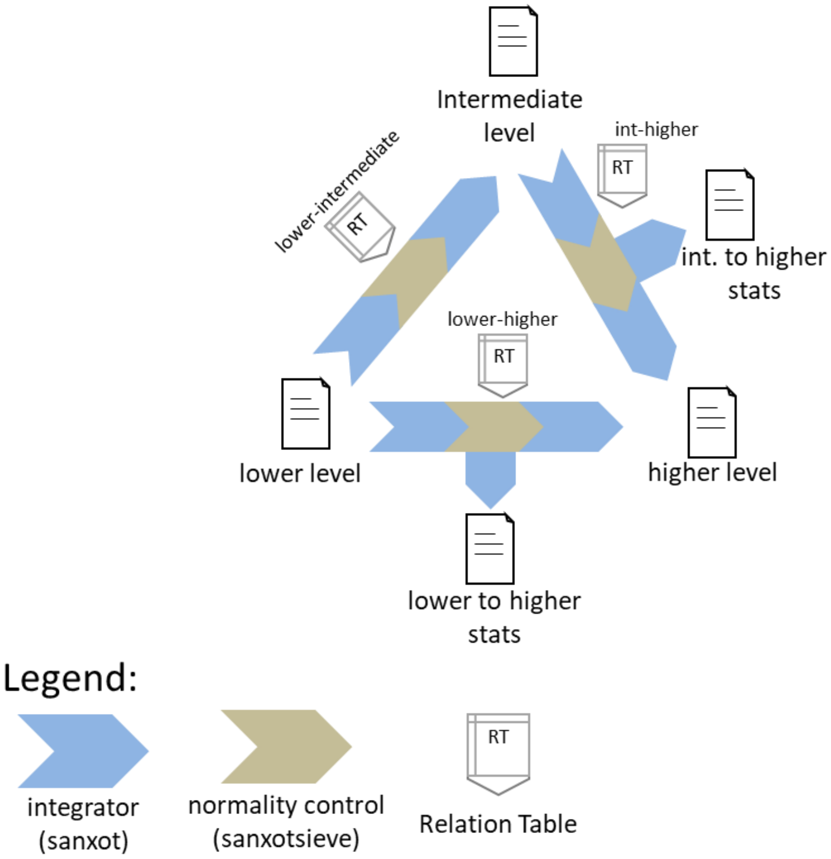


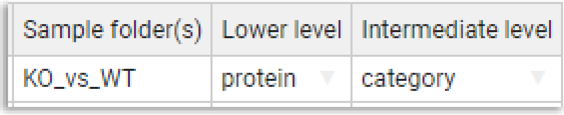
Figure 20. Schematic representation of the SBT module.

Standard parameters

The standard parameters to be entered in the SBT module task table are (**Figure 21**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Sample folder(s)* indicates the name of the folder(s) where the lower level data are located.
- *Lower level* indicates the name of the lower level (e.g. protein).
- *Intermediate level* indicates the name of the intermediate level (e.g. category).

By default, the higher level is the grand mean of the lower level elements.



Sample folder(s)	Lower level	Intermediate level
KO_vs_WT	protein	category

Figure 21. A sample task table in the SBT module.

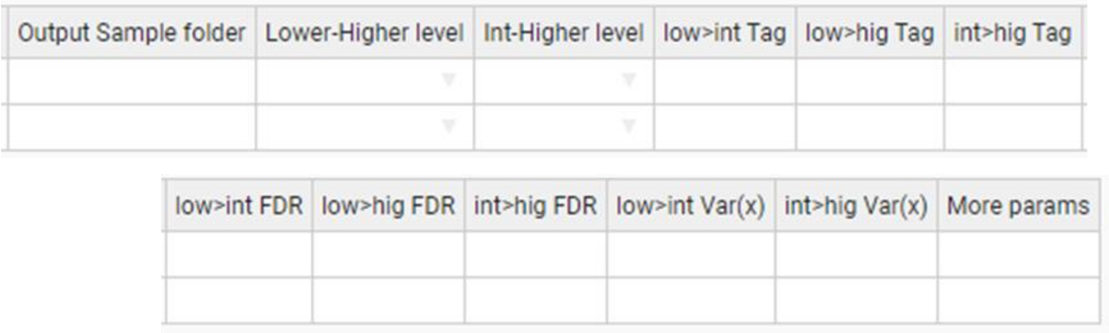
Advanced parameters

This module accepts the following additional parameters (**Figure 22**):

- *Output Sample folder* indicates an alternative folder to store the resulting log$_2$-ratio and statistical weight values other than “Sample folder(s)”.
- *Lower-Higher level* and *Int(ermediate)-Higher level* specify an alternative higher level other than the grand mean of the lower level elements.
- *Low(er)-to-Int(ermediate)* Tag and *Int(ermediate)-to-Hig(her)* are the text label that indicates which elements from the lower level are integrated into the intermediate level, and the intermediate level are integrated into higher level. The tags must be specified in the third column of the corresponding Relation Table. This allows the user to discard elements for integration without needing to eliminate them from the Relation Table. Thus, if the label “marked” is used as a Tag, only the lower level elements containing the label marked in the third column of the lower_level2intermediate_level and intermediate_level2higher_level Relation Tables will be integrated. Logical operators can also be used in the Tag field to make complex decisions.

  By default, iSanXoT eliminates outliers from the lower level according to an FDR<1% threshold.

  For further details, read the Advanced Parameters for the INTEGRATE module.
- *Low(er)-to-Int(ermediate) FDR* and *Int(ermediate)-to-Hig(her) FDR* determine an FDR threshold other than the default value (0.01, i.e. 1%) for outlier removal in the lower level-to-intermediate level and intermediate level-to-higher level integration, respectively. If “0” is specified as the FDR value, then no outliers will be discarded.
- *Low(er)-to-Int(ermediate) Var(x)* and *Int(ermediate)-to-Hig(her) Var(x)* indicate the variance to be used in the lower level-to-intermediate level and intermediate level-to-higher level integration, respectively, as an alternative to the variance calculated in the lower level-to-intermediate level integration.
- *More params* allows adding more parameters to the internal programs of the module. For more details see More params in the “Special Parameters” Section.



Output Sample folder	Lower-Higher level	Int-Higher level	low>int Tag	low>hig Tag	int>hig Tag
	▾	▾			
	▾	▾			

low>int FDR	low>hig FDR	int>hig FDR	low>int Var(x)	int>hig Var(x)	More params

Figure 22. A sample task table with advanced parameters in the SBT module.

Compound modules

The Compound modules perform a sequence of consecutive integrations based on the WSPP statistical model [2] and the SBT algorithm [3]. In addition, each module creates the initial level and calibrate this initial level. The WSPP-SBT and WSPPG-SBT modules create and calibrate the “scan” level; and the WPP-SBT and WPPG-SBT create and calibrate the “peptide” level.

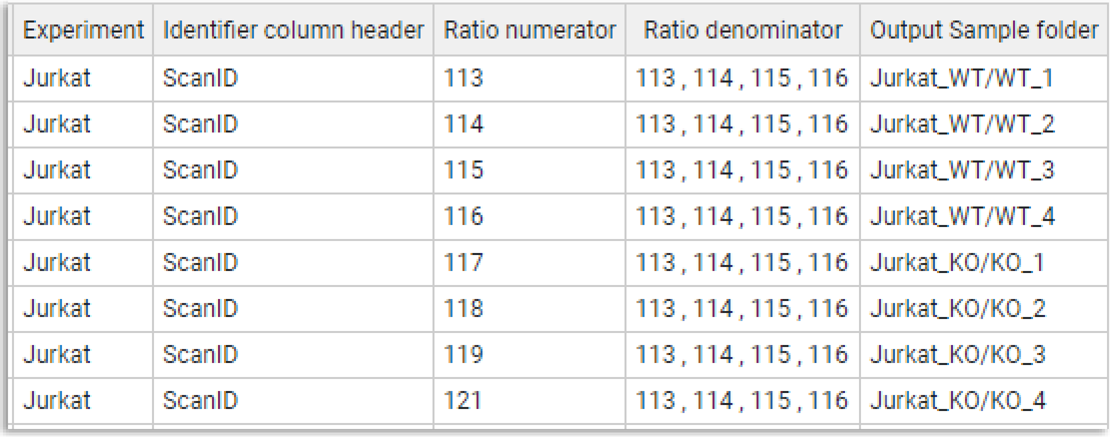
WSPP-SBT

The WSPP-SBT module performs the following integrations: scan-to-peptide, peptide-to-protein, protein-to-category, protein-to-proteinall, and category-to-categoryall. In addition, the SBT algorithm is used to calculate the variance associated to the protein-to-category integration, which is applied to the protein-to-proteinall integration.

Standard parameters

The standard parameters required by the WSPP-SBT module are (**Figure 23**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Experiment* is the column header that designates which elements from the Identification file will be used in the starting scan-to-peptide integration.
- *Identifier column header* is the identification file column header that unambiguously identifies the scans.
- *Ratio numerator* specifies which identification file column header designates the quantitative values to be used as a numerator for the $\log_2$-ratio calculation.
- *Ratio denominator* specifies which identification file column header designates the quantitative values to be used as a denominator for the $\log_2$-ratio calculation.
- Output Sample folder: indicates the name of the folder where the resulting data files will be saved.



Experiment	Identifier column header	Ratio numerator	Ratio denominator	Output Sample folder
Jurkat	ScanID	113	113 , 114 , 115 , 116	Jurkat_WT/WT_1
Jurkat	ScanID	114	113 , 114 , 115 , 116	Jurkat_WT/WT_2
Jurkat	ScanID	115	113 , 114 , 115 , 116	Jurkat_WT/WT_3
Jurkat	ScanID	116	113 , 114 , 115 , 116	Jurkat_WT/WT_4
Jurkat	ScanID	117	113 , 114 , 115 , 116	Jurkat_KO/KO_1
Jurkat	ScanID	118	113 , 114 , 115 , 116	Jurkat_KO/KO_2
Jurkat	ScanID	119	113 , 114 , 115 , 116	Jurkat_KO/KO_3
Jurkat	ScanID	121	113 , 114 , 115 , 116	Jurkat_KO/KO_4

Figure 23. A sample task table in the WSPP-SBT, WSPPG-SBT, WPP-SBT, and WPPG-SBT modules.

Advanced parameters

The WSPP-SBT module accepts the following advanced parameters (**Figure 24**):

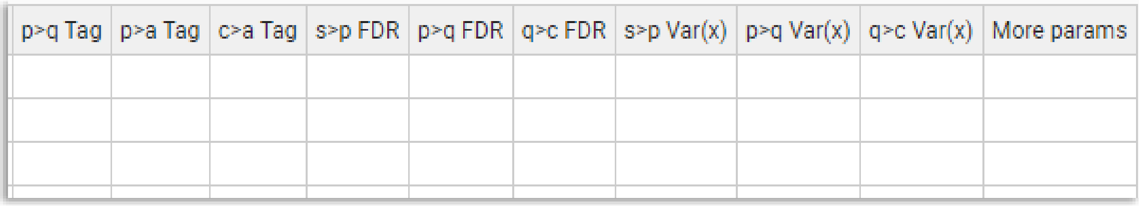
- *p>q Tag*, *p>a Tag*, *c>a Tag* are the text label that indicates which elements from the lower level ("p" and "c") are integrated into the higher level ("q" and "a"). The tags must be specified in the third column of the corresponding Relation Table. This allows the user to discard elements for integration without needing to eliminate them from the Relation Table. Thus, if the label "marked" is used as a Tag, only the lower level elements containing the label marked in the third column of the lower_level2higher_level

Relation table will be integrated. Logical operators can also be used in the Tag field to make complex decisions.

By default, iSanXoT eliminates outliers from the lower level according to an FDR<1% threshold.

For further details, read the Advanced Parameters for the INTEGRATE module.

- *s>p FDR*, *p>q FDR*, *q>c FDR* establish an FDR threshold other than 0.01 (1%) for outlier removal in the integrations scan-to-peptide, peptide-to-protein, and protein-to-category, respectively. If FDR = 0 is selected, then the outliers are not discarded.
- *s>p Var(x)*, *p>q Var(x)*, *q>c Var(x)* set a fixed value for the variance in the integrations scan-to-peptide, peptide-to-protein, and protein-to-category, respectively. By default, the variance will be iteratively calculated based on the Levenberg-Marquardt algorithm in the first lower level-to-higher level integration (**Figure 10**).
- *More params* allows adding more parameters to the internal programs of the module. For more details see More params in the "Special Parameters" Section.



p>q Tag	p>a Tag	c>a Tag	s>p FDR	p>q FDR	q>c FDR	s>p Var(x)	p>q Var(x)	q>c Var(x)	More params

Figure 24. Task-Table with advanced parameters in the WSPP-SBT module.

WSPPG-SBT

The WSPPG-SBT module performs the following integrations: scan-to-peptide, peptide-to-protein, protein-to-gene, gene-to-category, gene-to-geneall, and category-to-categoryall. In addition, the SBT algorithm is used to calculate the variance associated to the gen-to-category integration, which is applied to the gen-to-geneall integration.

Standard parameters

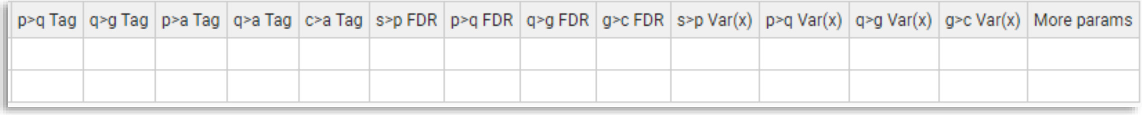
The standard parameters required by the WSPPG-SBT module are (**Figure 23**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Experiment* is the column header that designates which elements from the Identification file will be used in the starting scan-to-peptide integration.
- *Identifier column header* is the identification file column header that unambiguously identifies the scans.
- *Ratio numerator* specifies which identification file column header designates the quantitative values to be used as a numerator for the $\log_2$-ratio calculation.
- *Ratio denominator* specifies which identification file column header designates the quantitative values to be used as a denominator for the $\log_2$-ratio calculation.
- Output Sample folder: indicates the name of the folder where the resulting data files will be saved.

Advanced parameters

The WSPPG-SBT module accepts the following advanced parameters (**Figure 25**):

- *p>q Tag*, *q>g Tag*, *p>a Tag*, *c>a Tag* are the text label that indicates which elements from the lower level ("p", "q" and "c") are integrated into the higher level ("q", "g" and "a"). The tags must be specified in the third column of the corresponding Relation Table. This allows the user to discard elements for integration without needing to eliminate them from the Relation Table. Thus, if the label "marked" is used as a Tag, only the lower level elements containing the label marked in the third column of the lower_level2higher_level Relation table will be integrated. Logical operators can also be used in the Tag field to make complex decisions.
  By default, iSanXoT eliminates outliers from the lower level according to an FDR<1% threshold.
  For further details, read the Advanced Parameters for the INTEGRATE module.
- *s>p FDR*, *p>q FDR*, *q>g FDR*, *g>c FDR* establish an FDR threshold other than 0.01 (1%) for outlier removal in the following integrations: scan-to-peptide, peptide-to-protein, protein-to-gene, and gene-to-category, respectively. If FDR = 0 is selected, then the outliers are not discarded.
- *s>p Var(x)*, *p>q Var(x)*, *q>g Var(x)*, *g>c Var(x)* set a fixed value for the variance in the integrations: scan-to-peptide, peptide-to-protein, protein-to-gene, and gene-to-category, respectively. By default, the variance will be iteratively calculated based on the Levenberg-Marquardt algorithm in the first lower level-to-higher level integration (**Figure 10**).
- *More params* allows adding more parameters to the internal programs of the module. For more details see More params in the "Special Parameters" Section.



p>q Tag	q>g Tag	p>a Tag	q>a Tag	c>a Tag	s>p FDR	p>q FDR	q>g FDR	g>c FDR	s>p Var(x)	p>q Var(x)	q>g Var(x)	g>c Var(x)	More params

Figure 25. Task-table with advanced parameters in the WSPPG-SBT module.

WPP-SBT

The WPP-SBT module performs the integrations peptide-to-protein, protein-to-category, protein-to-proteinall and category-to-categoryall. In addition, the SBT algorithm is used to calculate the variance associated to the protein-to-category integration, which is applied to the protein-to-proteinall integration.

Standard parameters

The standard parameters required by the WPP-SBT module are (**Figure 23**):

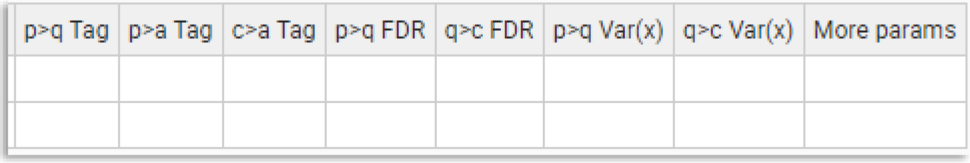
- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Experiment* is the column header that designates which elements from the Identification file will be used in the starting scan-to-peptide integration.
- *Identifier column header* is the identification file column header that unambiguously identifies the scans.

- *Ratio numerator* specifies which identification file column header designates the quantitative values to be used as a numerator for the log$_2$-ratio calculation.
- *Ratio denominator* specifies which identification file column header designates the quantitative values to be used as a denominator for the log$_2$-ratio calculation.
- Output Sample folder: indicates the name of the folder where the resulting data files will be saved.

Advanced parameters

The WPP-SBT module accepts the following advanced parameters (**Figure 26**):

- *p>q Tag*, *p>a Tag*, *c>a Tag* are the text label that indicates which elements from the lower level ("p" and "c") are integrated into the higher level ("q" and "a"). The tags must be specified in the third column of the corresponding Relation Table. This allows the user to discard elements for integration without needing to eliminate them from the Relation Table. Thus, if the label "marked" is used as a Tag, only the lower level elements containing the label marked in the third column of the lower_level2higher_level Relation Table will be integrated. Logical operators can also be used in the Tag field to make complex decisions.
  By default, iSanXoT eliminates outliers from the lower level according to an FDR<1% threshold.
  For further details, read the Advanced Parameters for the INTEGRATE module.
- *p>q FDR*, *q>c FDR* establish an FDR threshold other than 0.01 (1%) for outlier removal in the peptide-to-protein and protein-to-category integrations, respectively. If FDR = 0 is selected, then the outliers are not discarded.
- *p>q Var(x)*, *q>c Var(x)* set a fixed value for the variance in the peptide-to-protein and protein-to-category integrations, respectively. By default, the variance will be iteratively calculated based on the Levenberg-Marquardt algorithm in the first lower level-to-higher level integration (**Figure 10**).
- *More params* allows adding more parameters to the internal programs of the module. For more details see More params in the "Special Parameters" Section.



p>q Tag	p>a Tag	c>a Tag	p>q FDR	q>c FDR	p>q Var(x)	q>c Var(x)	More params

Figure 26. Task-table with advanced parameters in the WPP-SBT module.

WPPG-SBT

The WPPG-SBT module performs the integrations peptide-to-protein, protein-to-gene, gene-to-category, gene-to-geneall, and category-to-categoryall. In addition, the SBT algorithm is used to calculate the variance associated to the gene-to-category integration, which is applied to the gene-to-geneall integration.

Standard parameters

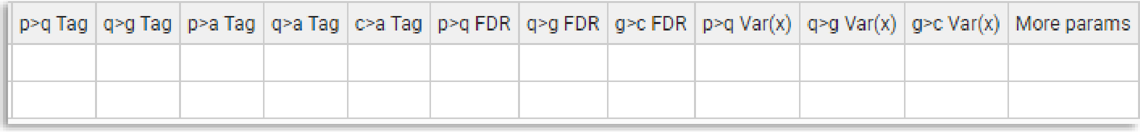
The standard parameters required by the WPPG-SBT module are (**Figure 23**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Experiment* is the column header that designates which elements from the Identification file will be used in the starting scan-to-peptide integration.
- *Identifier column header* is the identification file column header that unambiguously identifies the scans.
- *Ratio numerator* specifies which identification file column header designates the quantitative values to be used as a numerator for the log$_2$-ratio calculation.
- *Ratio denominator* specifies which identification file column header designates the quantitative values to be used as a denominator for the log$_2$-ratio calculation.
- Output Sample folder: indicates the name of the folder where the resulting data files will be saved.

Advanced parameters

The WPPG-SBT module accepts the following advanced parameters (**Figure 27**):

- *p>q Tag, q>g Tag, p>a Tag, q>a Tag, c>a Tag* are the text label that indicates which elements from the lower level ("p", "q" and "c") are integrated into the higher level ("q", "g" and "a"). The tags must be specified in the third column of the corresponding Relation Table. This allows the user to discard elements for integration without needing to eliminate them from the Relation Table. Thus, if the label "marked" is used as a Tag, only the lower level elements containing the label marked in the third column of the lower_level2higher_level Relation table will be integrated. Logical operators can also be used in the Tag field to make complex decisions.
  By default, iSanXoT eliminates outliers from the lower level according to an FDR<1% threshold.
  For further details, read the Advanced Parameters for the INTEGRATE module.
- *p>q FDR, q>g FDR, g>c FDR* establish an FDR threshold other than 0.01 (1%) for outlier removal in the integrations peptide-to-protein, protein-to-gene, and gene-to-category, respectively. If FDR = 0 is selected, then the outliers are not discarded.
- *p>q Var(x), q>g Var(x), g>c Var(x)* set a fixed value for the variance in the integrations peptide-to-protein, protein-to-gene, and gene-to-category, respectively. By default, the variance will be iteratively calculated based on the Levenberg-Marquardt algorithm in the first lower level-to-higher level integration (**Figure 10**).
- *More params* allows adding more parameters to the internal programs of the module. For more details see More params in the "Special Parameters" Section.



p>q Tag	q>g Tag	p>a Tag	q>a Tag	c>a Tag	p>q FDR	q>g FDR	g>c FDR	p>q Var(x)	q>g Var(x)	g>c Var(x)	More params

Figure 27. Task-table with advanced parameters in the WPPG-SBT module.

Reports modules

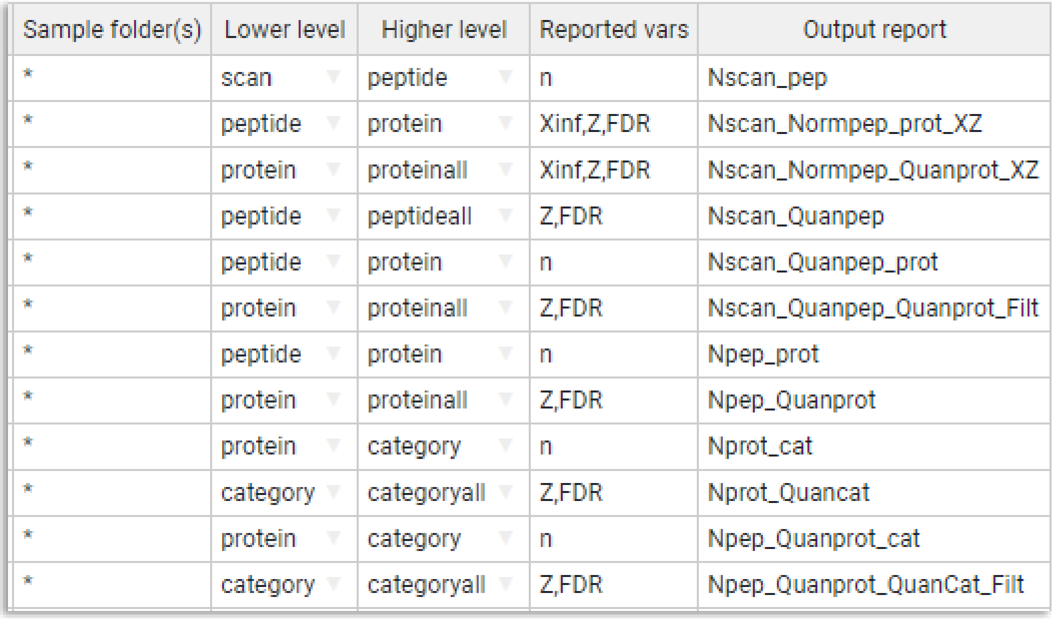
REPORT

The REPORT module allows the collection of the statistical variables (n, tags, Xinf, Vinf, Xsup, Vsup, Z, and FDR) from the different integrations performed into result tables.

Standard parameters

The standard parameters showing in the REPORT task table are (**Figure 28**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Sample folder(s)* indicates the name(s) of the folder(s) where the values of the statistical variables to be retrieved are located.
- *Lower level* indicates the starting level (i.e. peptide, protein, or category) for the integration whose statistical variables are to be reported.
- *Higher level* indicates the ending level for the integration whose statistical variables are to be reported.
- *Reported vars* specifies which statistical variables will be reported. The available variables are n, tags, Xinf, Vinf, Xsup, Vsup, Z, FDR, X'inf and Winf.
- *Output report* is the report filename (without extension).



Sample folder(s)	Lower level	Higher level	Reported vars	Output report
*	scan	peptide	n	Nscan_pep
*	peptide	protein	Xinf,Z,FDR	Nscan_Normpep_prot_XZ
*	protein	proteinall	Xinf,Z,FDR	Nscan_Normpep_Quanprot_XZ
*	peptide	peptideall	Z,FDR	Nscan_Quanpep
*	peptide	protein	n	Nscan_Quanpep_prot
*	protein	proteinall	Z,FDR	Nscan_Quanpep_Quanprot_Filt
*	peptide	protein	n	Npep_prot
*	protein	proteinall	Z,FDR	Npep_Quanprot
*	protein	category	n	Nprot_cat
*	category	categoryall	Z,FDR	Nprot_Quancat
*	protein	category	n	Npep_Quanprot_cat
*	category	categoryall	Z,FDR	Npep_Quanprot_QuanCat_Filt

Figure 28. A sample task table in the REPORT module.

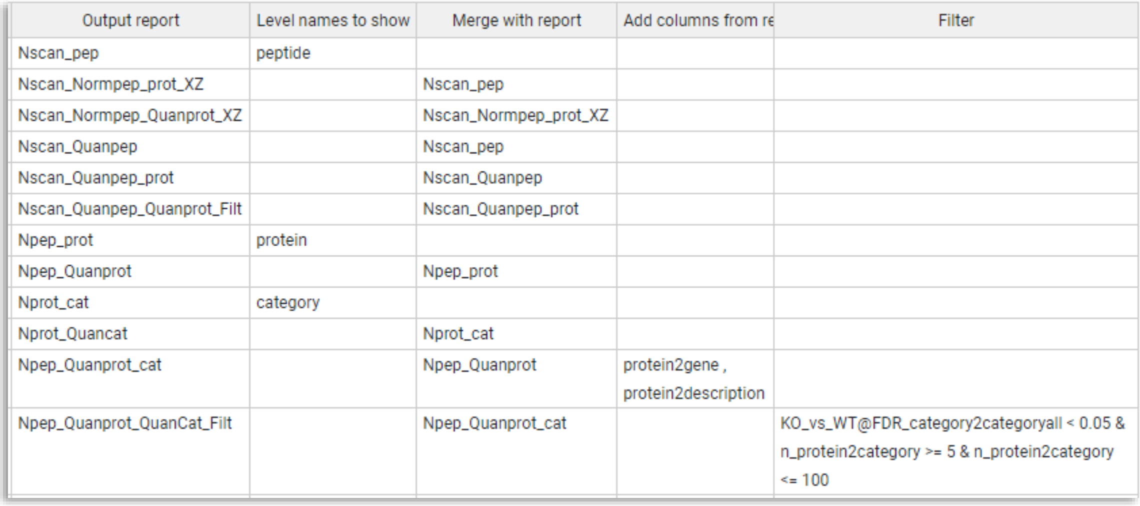
For instance, the first row of the task table shown in **Figure 28** prompts the REPORT module to read the variable "n" from the *scan2peptide_outStats.tsv* file that contains the statistical outcome from the scan-to-peptide integration (the asterisk wildcard character in *Sample* *folder(s)* causes REPORT to retrieve the "n" variable from every sample). These "n" values are written to a report file named "Nscan_pep" that is stored in the project "reports" folder.

The second row instructs the module to read the variables "Xinf", "Z" and "FDR" from the statistical outcome of the peptide-to-protein integration (once again for every sample). These values are written to a report file named "Nscan_Normpep_prot_XZ".

Advanced parameters

The REPORT module accepts the following advanced parameters (**Figure 29**):

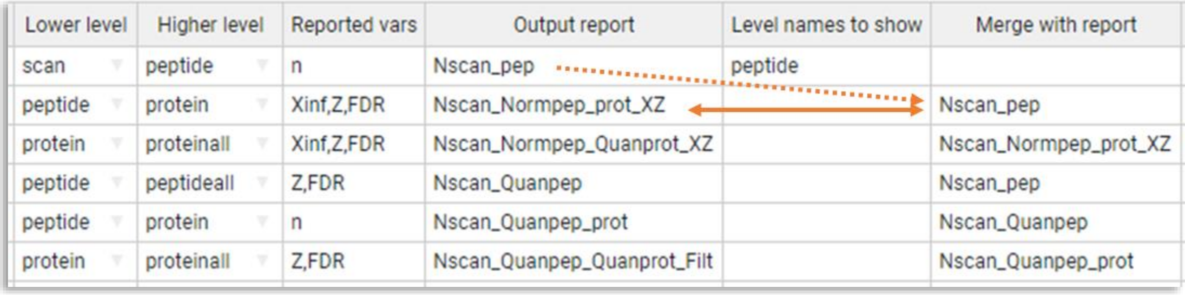
- *Level names to show* allows the user to restrict the elements to be written to the *Output report* to those from the *Lower level* or the *Higher level*. Both levels are used by default.
- *Merge with report* designates the file whose *Reported vars* will be incorporated into the *Output report* after intersection with the latter file.
- *Add columns from relation table* appends *Lower level* elements, extracted from the relation table designated, to the *Output report*. It is possible to indicate multiple relation tables separated by a comma.
- *Filter* allows to filter the data to be transferred to the *Output report* based on the *Reported vars* (n, Z, FDR, etc.). For more details, see Filter in the "Special Parameters" Section.



Output report	Level names to show	Merge with report	Add columns from re	Filter
Nscan_pep	peptide			
Nscan_Normpep_prot_XZ		Nscan_pep		
Nscan_Normpep_Quanprot_XZ		Nscan_Normpep_prot_XZ		
Nscan_Quanpep		Nscan_pep		
Nscan_Quanpep_prot		Nscan_Quanpep		
Nscan_Quanpep_Quanprot_Filt		Nscan_Quanpep_prot		
Npep_prot	protein			
Npep_Quanprot		Npep_prot		
Nprot_cat	category			
Nprot_Quancat		Nprot_cat		
Npep_Quanprot_cat		Npep_Quanprot	protein2gene , protein2description	
Npep_Quanprot_QuanCat_Filt		Npep_Quanprot_cat		KO_vs_WT@FDR_category2categoryall < 0.05 & n_protein2category >= 5 & n_protein2category <= 100

Figure 29. A sample task-table with advanced parameters in the REPORT module.

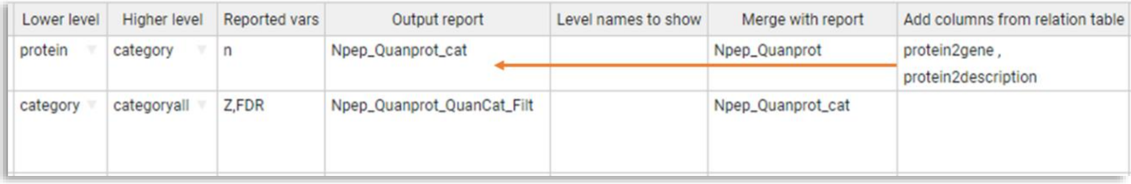
The reports indicated under *Output report* and *Merge with report* are merged according to the column header that they share. Thus, the REPORT task table shown in **Figure 30** will cause the module to incorporate the number of scans per peptide, displayed in the report "Nscan_pep", to the report "Nscan_Normpep_prot_XZ", as these two reports share the lower level elements showing under the "peptide" header.



Lower level	Higher level	Reported vars	Output report	Level names to show	Merge with report
scan	peptide	n	Nscan_pep	peptide	
peptide	protein	Xinf,Z,FDR	Nscan_Normpep_prot_XZ		Nscan_pep
protein	proteinall	Xinf,Z,FDR	Nscan_Normpep_Quanprot_XZ		Nscan_Normpep_prot_XZ
peptide	peptideall	Z,FDR	Nscan_Quanpep		Nscan_pep
peptide	protein	n	Nscan_Quanpep_prot		Nscan_Quanpep
protein	proteinall	Z,FDR	Nscan_Quanpep_Quanprot_Filt		Nscan_Quanpep_prot

Figure 30. Report merging in the REPORT module. *The first task table row creates a report file ("Nscan_pep") with the (n)umber of scans per peptide. The second row creates a report file called "Nscan_Normpep_prot_XZ" that contains, apart from the variables "Xinf", "Z", and "FDR" coming from the peptide-to-protein integration, the (n)umber of scans per peptide previously stored in the "Nscan_pep" report, as these two reports share the lower level elements showing under the "peptide" header.*

In addition, it is possible to incorporate additional data from one or more relation tables into the reports (**Figure 31**). When a given Relation Table is indicated under *Add columns from relation table*, the REPORT module will first attempt to incorporate to the *Output report* ("Npep_Quanprot_cat" in the example) the elements related to the *Lower level* ("protein" in this case) elements in the relation table. If the *Lower level* elements are missing in the relation table, then REPORT will try to incorporate the elements related to the Higher level ("category" in this case) instead. If neither the *Lower level* nor the *Higher level* can be found in the relation table, then no action is performed.



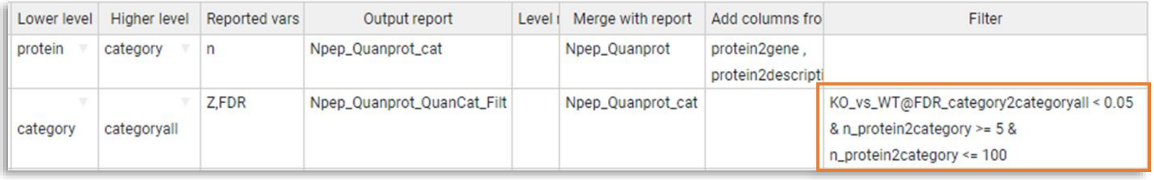
Lower level	Higher level	Reported vars	Output report	Level names to show	Merge with report	Add columns from relation table
protein	category	n	Npep_Quanprot_cat		Npep_Quanprot	protein2gene , protein2description
category	categoryall	Z,FDR	Npep_Quanprot_QuanCat_Filt		Npep_Quanprot_cat	

Figure 31. Adding data from relation tables into the report files. *The relation table "protein2gene" contains a column with protein identifiers under the "protein" header and another column with the corresponding gene name under the "gene" header, whereas the relation table "protein2description" contains, apart from the "protein" elements, a column with the corresponding protein description. The first task table row will prompt REPORT to incorporate the gene names and protein descriptions contained in the relation tables to the report file "Npep_Quanprot_cat" report file, as "protein" is the* Lower level *they all three shares.*

Finally, the report data can be filtered performing logical operations with the *Reported vars* in the *Filter* field. For instance, in the report task table displayed in **Figure 32**:

- n_protein2category <= 100, filters out from the report the variables Z and FDR for the category-to-categoryall integration when the (n)umber of proteins per category is greater than 100.

- n_protein2category >= 5 & n_protein2category <= 100 retrieves the variables Z and FDR for the category-to-categoryall integration when the (n)umber of proteins per category is in the [5, 100] range.
- KO_vs_WT@FDR_category2categoryall < 0.05, retrieves the variables Z and FDR for the category-to-categoryall integration provided that the FDR corresponding to the "KO_vs_WT" samples is less than 0.05.



Lower level	Higher level	Reported vars	Output report	Level i	Merge with report	Add columns fro	Filter
protein	category	n	Npep_Quanprot_cat		Npep_Quanprot	protein2gene , protein2descripti	
category	categoryall	Z,FDR	Npep_Quanprot_QuanCat_Filt		Npep_Quanprot_cat		KO_vs_WT@FDR_category2categoryall < 0.05 & n_protein2category >= 5 & n_protein2category <= 100

Figure 32. Filtering the report data.

The compound variables shown in the *Filter* field on **Figure 32** follow the structure *Reported var_integration*, like "n_protein2category". Such filter applies to all samples.

However, the filter "KO_vs_WT@FDR_category2categoryall" is applied based on the variable "FDR" from the category-to-categoryall integration, but only to the "KO_vs_WT" sample. Moreover, the filter "WT1,WT2@FDR_category2categoryall" is applied to the "WT1" and "WT2" samples.

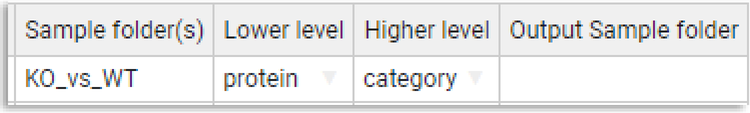
SANSON

The SANSON module generates a similarity graph showing the relationship between functional categories based on their protein components.

Standard parameters

The standard parameters to be provided for this module are (**Figure 33**):

- *Forced execution*: This checkbox field indicates whether to force the execution or not.
- *Sample folder(s)* indicates the names of the folder(s) where the lower level data file is located.
- *Lower level* indicates the name of the lower level elements ("protein" in this case) to be used.
- *Higher level* indicates the name of the higher level ("category" in this case) to which the lower level elements will be integrated.
- *Output Sample folder* designates a folder other than *Sample folder(s)* where the results will be saved.



Sample folder(s)	Lower level	Higher level	Output Sample folder
KO_vs_WT	protein	category	

Figure 33. A sample task table in the SANSON module.

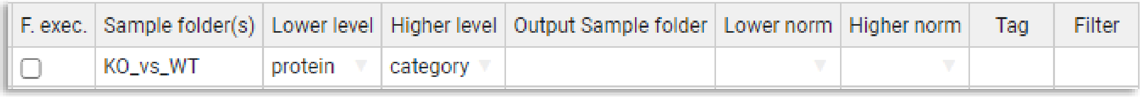
Advanced parameters

The SANSON module accepts the following additional parameters (**Figure 34**):

- *Lower norm* specifies the normalization scheme to be used with the lower level elements. The default value is included the normalization of lower level to all.
- *Higher norm* specifies the normalization scheme to be used with the higher level elements. The default value is included the normalization of higher level to all.
- *Tag* is a text label that indicates which elements from the lower level are integrated into the higher level. The tags must be specified in the third column of the corresponding Relation Table. This allows the user to discard elements for integration without needing to eliminate them from the Relation Table. Thus, if the label "marked" is used as a Tag, only the lower level elements containing the label marked in the third column of the lower_level2higher_level Relation table will be integrated. Logical operators can also be used in the Tag field to make complex decisions.

  By default, iSanXoT eliminates outliers from the lower level according to an FDR<1% threshold.

  For further details, read the Advanced Parameters for the INTEGRATE module.
- *Filter* allows to filter the data based on the FDR and number of proteins. For more details, see Filter in the "Special Parameters" Section.



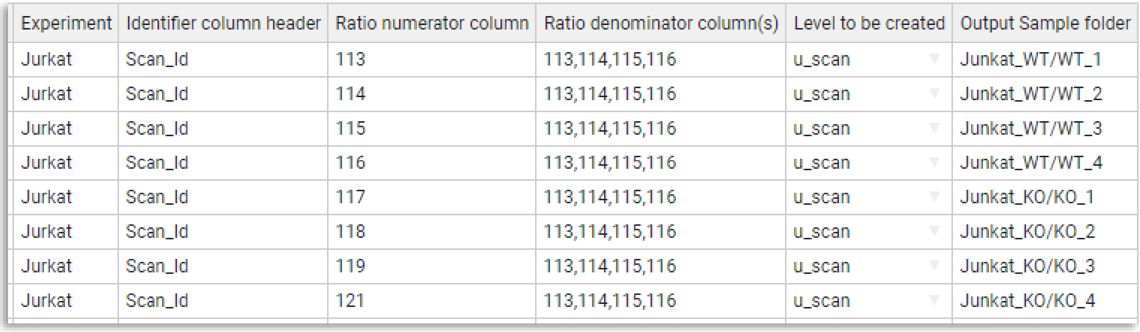
F. exec.	Sample folder(s)	Lower level	Higher level	Output Sample folder	Lower norm	Higher norm	Tag	Filter
☐	KO_vs_WT	protein	category					

Figure 34. A sample task-table with advanced parameters in the SANSON module.

Special parameters

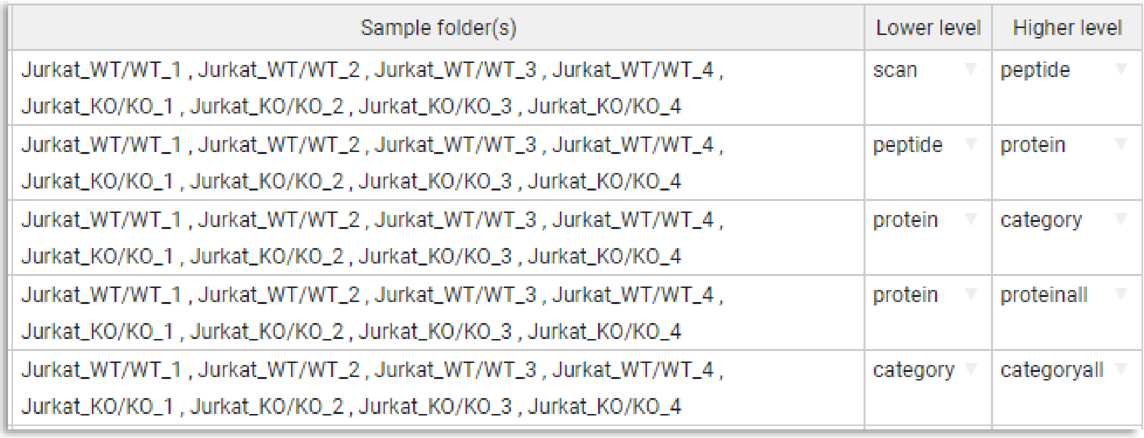
Multiple samples

The "Sample folder(s)" field of the different module task tables admit multiple samples. For instance, let's consider the samples created with the following LEVEL CREATOR task table:



Experiment	Identifier column header	Ratio numerator column	Ratio denominator column(s)	Level to be created	Output Sample folder
Jurkat	Scan_Id	113	113,114,115,116	u_scan	Junkat_WT/WT_1
Jurkat	Scan_Id	114	113,114,115,116	u_scan	Junkat_WT/WT_2
Jurkat	Scan_Id	115	113,114,115,116	u_scan	Junkat_WT/WT_3
Jurkat	Scan_Id	116	113,114,115,116	u_scan	Junkat_WT/WT_4
Jurkat	Scan_Id	117	113,114,115,116	u_scan	Junkat_KO/KO_1
Jurkat	Scan_Id	118	113,114,115,116	u_scan	Junkat_KO/KO_2
Jurkat	Scan_Id	119	113,114,115,116	u_scan	Junkat_KO/KO_3
Jurkat	Scan_Id	121	113,114,115,116	u_scan	Junkat_KO/KO_4

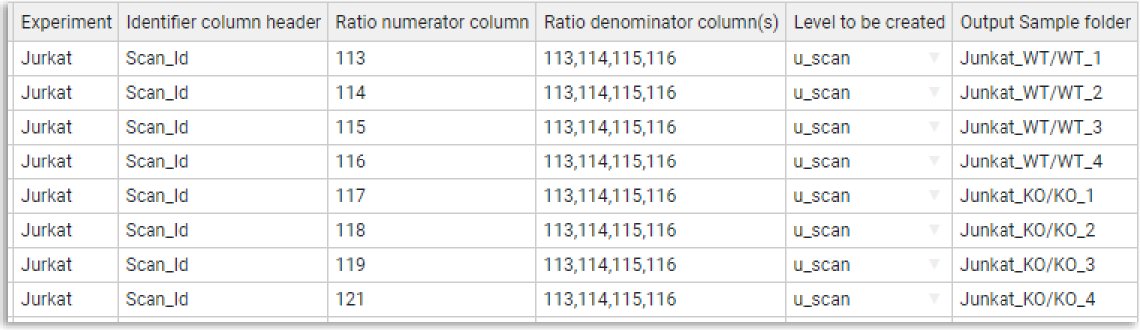
We can include multiple samples separated by a comma, for example, in the INTEGRATE module task table:



Sample folder(s)	Lower level	Higher level
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	scan	peptide
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	peptide	protein
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	protein	category
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	protein	proteinall
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	category	categoryall

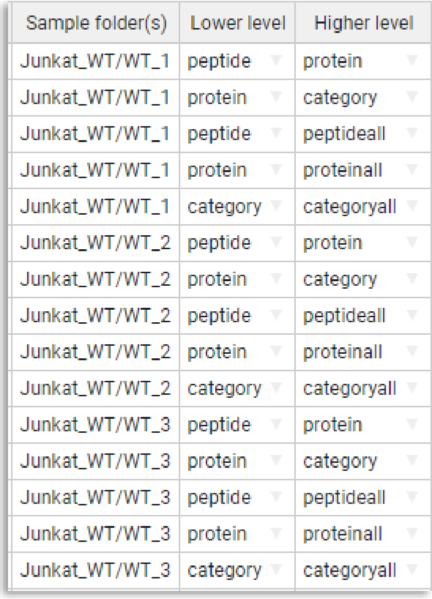
Asterisk is our jack of all trades

The module task tables admit the usage of the asterisk symbol as a wildcard character. Let's once more consider the samples created with the following LEVEL CREATOR task table:



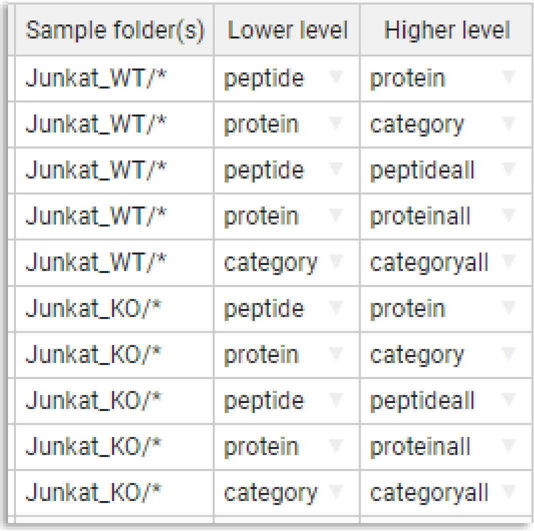
Experiment	Identifier column header	Ratio numerator column	Ratio denominator column(s)	Level to be created	Output Sample folder
Jurkat	Scan_Id	113	113,114,115,116	u_scan	Jurkat_WT/WT_1
Jurkat	Scan_Id	114	113,114,115,116	u_scan	Jurkat_WT/WT_2
Jurkat	Scan_Id	115	113,114,115,116	u_scan	Jurkat_WT/WT_3
Jurkat	Scan_Id	116	113,114,115,116	u_scan	Jurkat_WT/WT_4
Jurkat	Scan_Id	117	113,114,115,116	u_scan	Jurkat_KO/KO_1
Jurkat	Scan_Id	118	113,114,115,116	u_scan	Jurkat_KO/KO_2
Jurkat	Scan_Id	119	113,114,115,116	u_scan	Jurkat_KO/KO_3
Jurkat	Scan_Id	121	113,114,115,116	u_scan	Jurkat_KO/KO_4

Each row calculates a ratio that is saved to the corresponding *Output Sample folder*. Thus, the ratio of 113 to the mean of 113, 114, 115, and 116 is saved to the "Jurkat_WT/WT_1" folder; the 114 to the mean of 113, 114, 115 ratios is saved to "Jurkat_WT/WT_2", and so on. One way to create the task table of the INTEGRATE module could be the following, where each row represents an integration for a given sample:



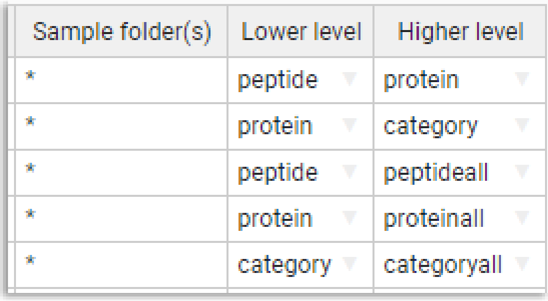
Sample folder(s)	Lower level	Higher level
Junkat_WT/WT_1	peptide	protein
Junkat_WT/WT_1	protein	category
Junkat_WT/WT_1	peptide	peptideall
Junkat_WT/WT_1	protein	proteinall
Junkat_WT/WT_1	category	categoryall
Junkat_WT/WT_2	peptide	protein
Junkat_WT/WT_2	protein	category
Junkat_WT/WT_2	peptide	peptideall
Junkat_WT/WT_2	protein	proteinall
Junkat_WT/WT_2	category	categoryall
Junkat_WT/WT_3	peptide	protein
Junkat_WT/WT_3	protein	category
Junkat_WT/WT_3	peptide	peptideall
Junkat_WT/WT_3	protein	proteinall
Junkat_WT/WT_3	category	categoryall

However, this task table can be simplified applying the asterisk wildcard. For instance, the task-table below allows to indicate multiple sample folders, namely every folder starting with "Jurkat_WT/" or "Jurkat_KO/".



Sample folder(s)	Lower level	Higher level
Junkat_WT/*	peptide	protein
Junkat_WT/*	protein	category
Junkat_WT/*	peptide	peptideall
Junkat_WT/*	protein	proteinall
Junkat_WT/*	category	categoryall
Junkat_KO/*	peptide	protein
Junkat_KO/*	protein	category
Junkat_KO/*	peptide	peptideall
Junkat_KO/*	protein	proteinall
Junkat_KO/*	category	categoryall

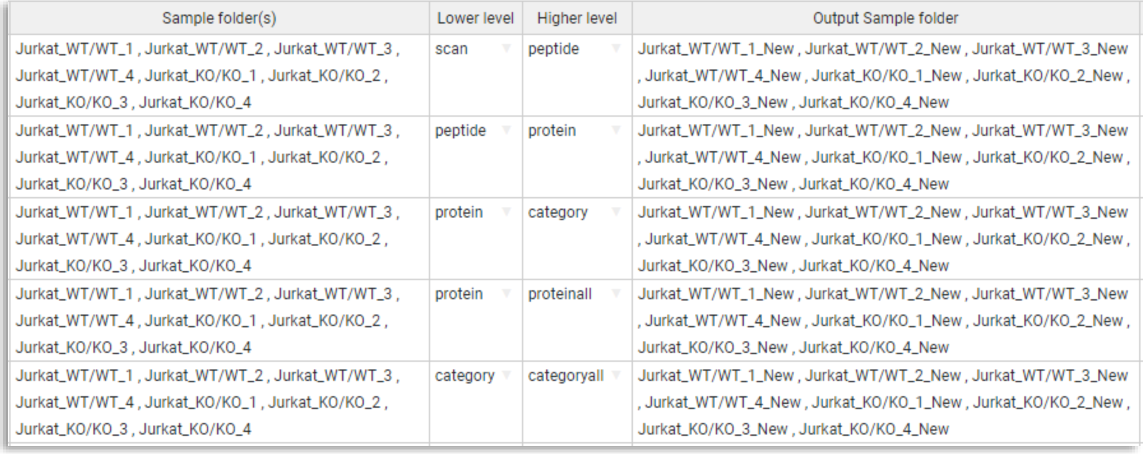
We can reduce this expression even more using just an asterisk: the first row of the following task table performs the integrations peptide-to-protein, protein-to-category, peptide-to-peptideall, protein-to-proteinall, and category-to-categoryall in every sample folder defined with LEVEL CREATOR.



Sample folder(s)	Lower level	Higher level
*	peptide	protein
*	protein	category
*	peptide	peptideall
*	protein	proteinall
*	category	categoryall

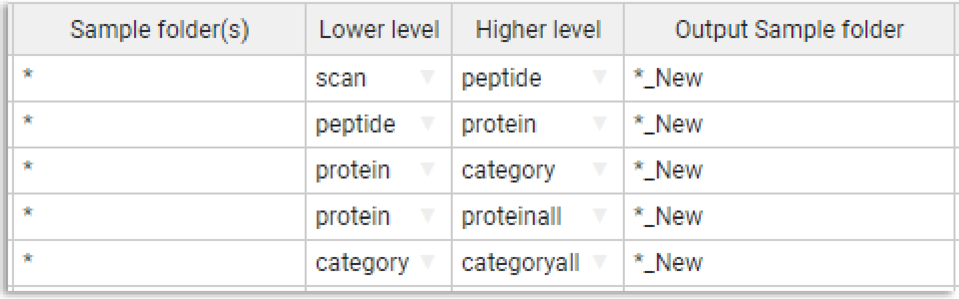
Multiple samples in the inputs and outputs

In the cases we have multiple input samples separated by comma but we want to save the results in another output sample folder, we indicate them in the same way giving the output folders separated by comma. It is required to provide the same number of folders:



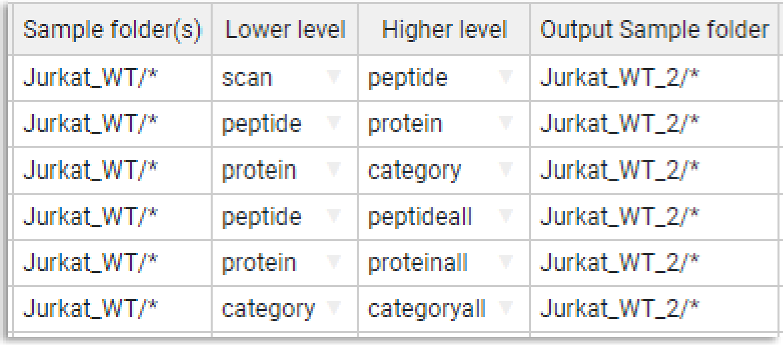
Sample folder(s)	Lower level	Higher level	Output Sample folder
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	scan	peptide	Jurkat_WT/WT_1_New , Jurkat_WT/WT_2_New , Jurkat_WT/WT_3_New , Jurkat_WT/WT_4_New , Jurkat_KO/KO_1_New , Jurkat_KO/KO_2_New , Jurkat_KO/KO_3_New , Jurkat_KO/KO_4_New
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	peptide	protein	Jurkat_WT/WT_1_New , Jurkat_WT/WT_2_New , Jurkat_WT/WT_3_New , Jurkat_WT/WT_4_New , Jurkat_KO/KO_1_New , Jurkat_KO/KO_2_New , Jurkat_KO/KO_3_New , Jurkat_KO/KO_4_New
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	protein	category	Jurkat_WT/WT_1_New , Jurkat_WT/WT_2_New , Jurkat_WT/WT_3_New , Jurkat_WT/WT_4_New , Jurkat_KO/KO_1_New , Jurkat_KO/KO_2_New , Jurkat_KO/KO_3_New , Jurkat_KO/KO_4_New
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	protein	proteinall	Jurkat_WT/WT_1_New , Jurkat_WT/WT_2_New , Jurkat_WT/WT_3_New , Jurkat_WT/WT_4_New , Jurkat_KO/KO_1_New , Jurkat_KO/KO_2_New , Jurkat_KO/KO_3_New , Jurkat_KO/KO_4_New
Jurkat_WT/WT_1 , Jurkat_WT/WT_2 , Jurkat_WT/WT_3 , Jurkat_WT/WT_4 , Jurkat_KO/KO_1 , Jurkat_KO/KO_2 , Jurkat_KO/KO_3 , Jurkat_KO/KO_4	category	categoryall	Jurkat_WT/WT_1_New , Jurkat_WT/WT_2_New , Jurkat_WT/WT_3_New , Jurkat_WT/WT_4_New , Jurkat_KO/KO_1_New , Jurkat_KO/KO_2_New , Jurkat_KO/KO_3_New , Jurkat_KO/KO_4_New

In the same way happens with the asterisk character (jack of all trades). In the “Output Sample folder(s)” we can add a suffix in the input samples. The following task-table illustrate that the output sample folders would be contain the “_New” suffix:



Sample folder(s)	Lower level	Higher level	Output Sample folder
*	scan	peptide	*_New
*	peptide	protein	*_New
*	protein	category	*_New
*	protein	proteinall	*_New
*	category	categoryall	*_New

In addition, we can rename the subfolder adding the new name in the “Output Sample folder(s)” or add a new subfolder:



Sample folder(s)	Lower level	Higher level	Output Sample folder
Jurkat_WT/*	scan	peptide	Jurkat_WT_2/*
Jurkat_WT/*	peptide	protein	Jurkat_WT_2/*
Jurkat_WT/*	protein	category	Jurkat_WT_2/*
Jurkat_WT/*	peptide	peptideall	Jurkat_WT_2/*
Jurkat_WT/*	protein	proteinall	Jurkat_WT_2/*
Jurkat_WT/*	category	categoryall	Jurkat_WT_2/*

More params

Some modules accept a column parameter in the Task-Table called "More params". This column allows you to provide advanced parameters for the SanXoT programs [1]. The program descriptions are in the following wiki link:

https://www.cnic.es/wiki/proteomica/index.php/SanXoT_software_package

The iSanXoT module are composed by several programs of SanXoT. For this reason, the "More params" of a module accepts the advanced parameters of composed programs indicated by a name.

For example,

INTEGRATE:

```
"sanxot1": " -m 300 -g ", "sanxot2": "-s --sweepdecimals=2.5"
```

In above example the first "sanxot" program that compose the INTEGRATE module, receives the "-m 300 -g" as parameter, and the second "sanxot" receives also the parameter "-s --sweepdecimals=2.5".

WSPP-SBT:

```
"p2q_sanxot2": " -m 100 -s ", "q2a_sanxot1": "-m 100"
```

In this example, the WSPP-SBT module has multiple integrations: scan-to-peptide, peptide-to-protein, etc. For more information, see the WSPP-SBT section. Thus, the second "sanxot" program of peptide-to-protein (q) integration will receive the parameter "-m 100 -s", and then, the first "sanxot" of protein (q)-to-proteinall (a) integration will receive the "-m 100".

The program names for each Module

INTEGRATE: sanxot1, sanxotsieve, sanxot2

NORCOMBINE: create_exp_tags, cardenio, sanxot1, sanxotsieve, sanxot2

SBT: l2i_sanxot1, l2i_sanxotsieve, l2i_sanxot2, i2h_sanxot1, i2h_sanxotsieve, i2h_sanxot2, l2h_sanxot1, l2h_sanxotsieve, l2h_sanxot2

WSPP_SBT: level_creator, klibrate, s2p_sanxot1, s2p_sanxotsieve, s2p_sanxot2, p2q_sanxot1, p2q_sanxotsieve, p2q_sanxot2, q2c_sanxot1, q2c_sanxotsieve, q2c_sanxot2, p2a_sanxot1, p2a_sanxotsieve, p2a_sanxot2, q2a_sanxot1, q2a_sanxotsieve, q2a_sanxot2, c2a_sanxot1, c2a_sanxotsieve, c2a_sanxot2

WSPPG_SBT: level_creator, klibrate, s2p_sanxot1, s2p_sanxotsieve, s2p_sanxot2, p2q_sanxot1, p2q_sanxotsieve, p2q_sanxot2, p2g_sanxot1, p2g_sanxotsieve, p2g_sanxot2, q2g_sanxot1, q2g_sanxotsieve, q2g_sanxot2, g2c_sanxot1, g2c_sanxotsieve, g2c_sanxot2, p2a_sanxot1, p2a_sanxotsieve, p2a_sanxot2, q2a_sanxot1, q2a_sanxotsieve, q2a_sanxot2, g2a_sanxot1, g2a_sanxotsieve, g2a_sanxot2, c2a_sanxot1, c2a_sanxotsieve, c2a_sanxot2

WPP_SBT: level_creator, klibrate, p2q_sanxot1, p2q_sanxotsieve, p2q_sanxot2, q2c_sanxot1, q2c_sanxotsieve, q2c_sanxot2, p2a_sanxot1, p2a_sanxotsieve, p2a_sanxot2, q2a_sanxot1, q2a_sanxotsieve, q2a_sanxot2, c2a_sanxot1, c2a_sanxotsieve, c2a_sanxot2

WPPG_SBT: level_creator, klibrate, p2q_sanxot1, p2q_sanxotsieve, p2q_sanxot2, p2g_sanxot1, p2g_sanxotsieve, p2g_sanxot2, q2g_sanxot1, q2g_sanxotsieve, q2g_sanxot2, g2c_sanxot1, g2c_sanxotsieve, g2c_sanxot2, p2a_sanxot1, p2a_sanxotsieve, p2a_sanxot2, q2a_sanxot1, q2a_sanxotsieve, q2a_sanxot2, g2a_sanxot1, g2a_sanxotsieve, g2a_sanxot2, c2a_sanxot1, c2a_sanxotsieve, c2a_sanxot2

Filter (for REPORT and SANSON)

The REPORT module and SANSON module accept a Filter parameter. This parameter filters the data based on some variables depending on the module.

In the case of REPORT module, the filtered variables are the Reported vars: n, Z, FDR, etc. For instance:

FDR_category2category < 0.05) & (n_protein2category >=5) & (n_protein2category <= 100)

For the SANSON module, the filtered variables are FDR and the related number (n_rel). For example:

([FDR] < 0.05) & ([n_rel] >= 10) & ([n_rel] <= 100)

Different variables can be combined using the comparisons: >=, <=, !=, <>, ==, >, <; and using logical operators “and” (&), “or” (||), and “not” (!).

Sample Workflows with Application to Case Studies

We describe below in detail four sample workflows that illustrate the capacity of iSanXoT to statistically ascertain abundance changes in both multiplexed, isotopically labeled [3-5] and label-free [6] proteomics experiments.

Workflow 1: One-step quantification in a labeled experiment

Experimental

The identification and quantification data from García-Marqués *et al*. [3] were used to illustrate workflow 1 performance. This study characterizes the molecular alterations that take place along time when vascular smooth muscle cells (VSMCs) are treated with angiotensin-II (AngII). Cells were incubated with AngII for 0, 2, 4, 6, 8, and 10 h; after protein extraction and digestion, the resulting peptides were isobarically labelled with iTRAQ 8-plex reagents, mixed appropriately and analyzed by liquid chromatography-tandem mass spectrometry (LC-MS/MS). Protein identification was achieved by database searching against target and decoy sequences using Proteome Discoverer 1.4 (Thermo Fisher Scientific; see Section 3 below to learn how these data are adapted to be used with iSanXoT). The false discovery rate (FDR) for peptide identification was calculated using the refined method [7].

Workflow operation

Workflow 1 is made up of the compound module WSPP-SBT and the basic module REPORT (*Figure 35*A). The WSPP-SBT performs a sequence of consecutive integrations based on the WSPP statistical model [2] and the SBT algorithm [3].

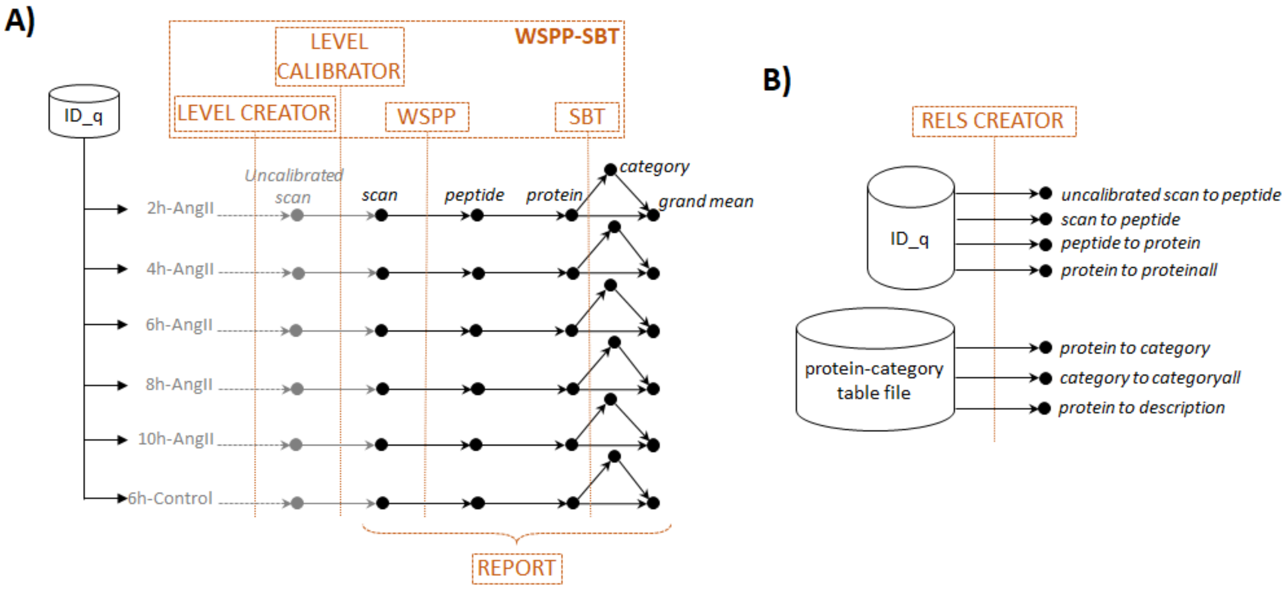
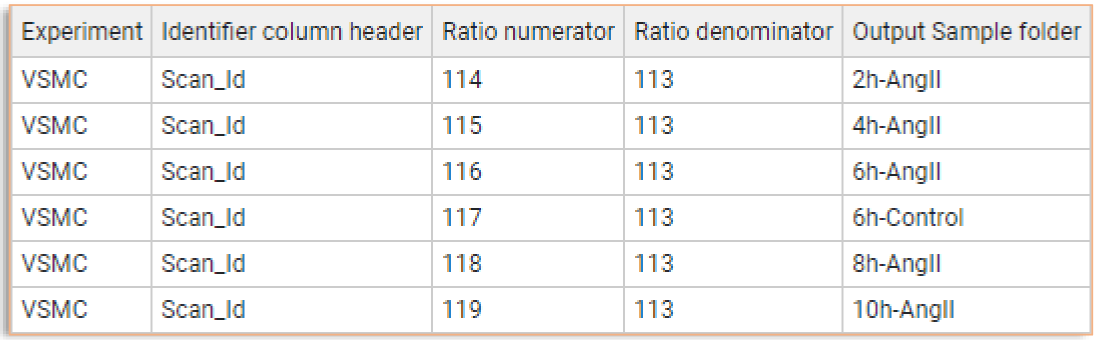


Figure 35. Scheme of workflow 1 (one-step quantification in a labeled experiment) showing module components: WSPP-SBT and REPORT (A) and RELS CREATOR (B).

The WSPP-SBT module starts with the creation of the log2 ratios at the scan (or spectrum) level. These log2 ratios are calculated using the numerator and denominator values indicated in the module task table (*Figure 36*) and subsequently calibrated to account for the variability associated with LC-MS/MS measurements, after which the *scan-to-peptide* and *peptide-to-protein* integrations are carried out.

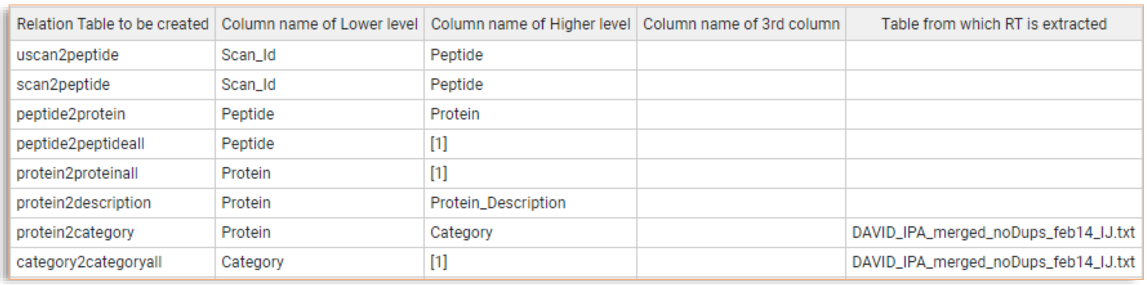


Experiment	Identifier column header	Ratio numerator	Ratio denominator	Output Sample folder
VSMC	Scan_Id	114	113	2h-AngII
VSMC	Scan_Id	115	113	4h-AngII
VSMC	Scan_Id	116	113	6h-AngII
VSMC	Scan_Id	117	113	6h-Control
VSMC	Scan_Id	118	113	8h-AngII
VSMC	Scan_Id	119	113	10h-AngII

Figure 36. The WSPP-SBT task table for workflow 1.

Then the SBT (systems biology triangle) algorithm is applied for the detection of functional category changes originated by the coordinated behaviour of proteins (*Figure 35*A). For that, the algorithm first calculates the variance of the *protein-to-category* integration, which is then applied to the *protein-to-protein grand mean* (hereinafter referred to as *protein-to-proteinall*) integration, and finally performs the *category-to-category grand mean* (hereinafter referred to as *category-to-categoryall*) integration. All these integrations created by the WSPP-SBT module are saved in the indicated *Output Sample folder*.

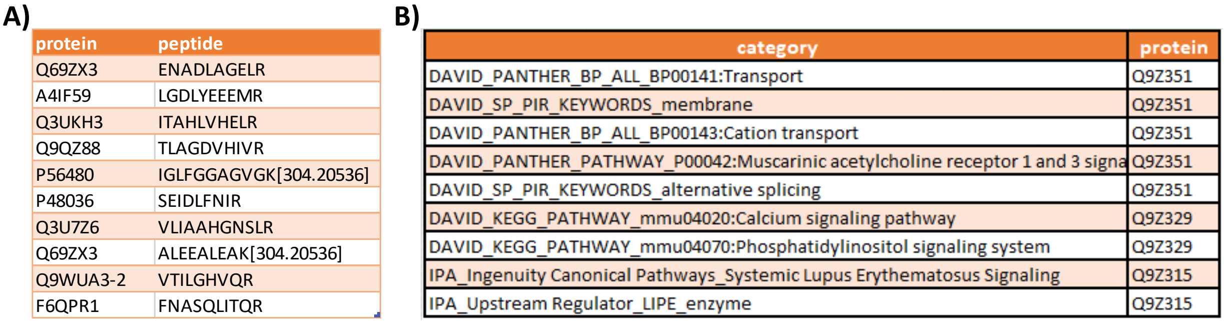
Every integration step needs a relation table that links lower- to higher-level elements. This text file is created by the RELS CREATOR module (*Figure 35*B and *Figure 37*).



Relation Table to be created	Column name of Lower level	Column name of Higher level	Column name of 3rd column	Table from which RT is extracted
uscan2peptide	Scan_Id	Peptide		
scan2peptide	Scan_Id	Peptide		
peptide2protein	Peptide	Protein		
peptide2peptideall	Peptide	[1]		
protein2proteinall	Protein	[1]		
protein2description	Protein	Protein_Description		
protein2category	Protein	Category		DAVID_IPA_merged_noDups_feb14_IJ.txt
category2categoryall	Category	[1]		DAVID_IPA_merged_noDups_feb14_IJ.txt

Figure 37. The RELS CREATOR task table for workflow 1.

Thus, the *peptide2protein* relation table (*Figure 38*A) is a text file linking the peptides identified to the proteins they come from. The elements of the *protein2category* relation table (*Figure 38*B) were retrieved from a text file containing functional annotations for mouse proteins gathered from several protein function databases (*Figure 37*), as described by the authors [3].



A)

protein	peptide
Q69ZX3	ENADLAGELR
A4IF59	LGDLYEEEMR
Q3UKH3	ITAHLVHELR
Q9QZ88	TLAGDVHIVR
P56480	IGLFGGAGVGK[304.20536]
P48036	SEIDLFNIR
Q3U7Z6	VLIAAHGNSLR
Q69ZX3	ALEEALEAK[304.20536]
Q9WUA3-2	VTILGHVQR
F6QPR1	FNASQLITQR

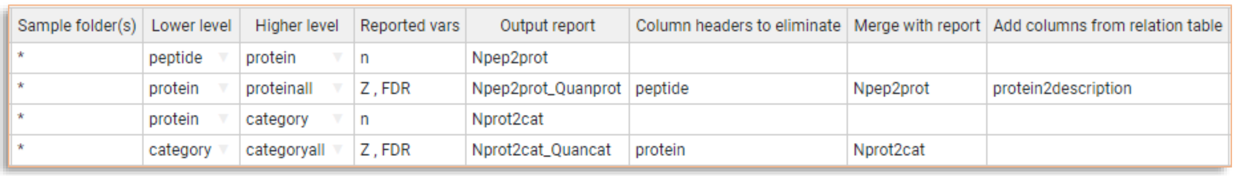
B)

category	protein
DAVID_PANTHER_BP_ALL_BP00141:Transport	Q9Z351
DAVID_SP_PIR_KEYWORDS_membrane	Q9Z351
DAVID_PANTHER_BP_ALL_BP00143:Cation transport	Q9Z351
DAVID_PANTHER_PATHWAY_P00042:Muscarinic acetylcholine receptor 1 and 3 signa	Q9Z351
DAVID_SP_PIR_KEYWORDS_alternative splicing	Q9Z351
DAVID_KEGG_PATHWAY_mmu04020:Calcium signaling pathway	Q9Z329
DAVID_KEGG_PATHWAY_mmu04070:Phosphatidylinositol signaling system	Q9Z329
IPA_Ingenuity Canonical Pathways_Systemic Lupus Erythematosus Signaling	Q9Z315
IPA_Upstream Regulator_LIPE_enzyme	Q9Z315

Figure 38. Excerpt from the peptide2protein (A) and protein2category (B) relation tables that link peptides to proteins and proteins to categories, respectively.

The REPORT module is used in this workflow to collect the following statistical variables for all samples (2h-AngII, 4h-AngII, 6h-AngII, 6h-Control, 8h-AngII, and10h-AngII), as prompted by the asterisk (*Figure 39*). The asterisk symbol is used in the task tables as a wildcard character. For further details, consult the iSanXoT wiki (https://github.com/CNIC-Proteomics/iSanXoT/wiki). The simple options used in this workflow are the following:

- Number of peptides per protein (*n*), obtained from the *peptide-to-protein* integration;
- Mean-corrected protein log2 ratios expressed in units of standard deviation (*Z*), calculated from the *protein-to-proteinall* integration;
- False Discovery Rate (*FDR*) values indicating which *Z* values stand out as outliers in the *protein-to-proteinall* integration.
- Number of proteins per category (*n*), obtained from the *protein-to-category* integration;
- Mean-corrected category log2 ratios expressed in units of standard deviation (*Z*), calculated from the *category-to-categoryall* integration;
- False Discovery Rate (*FDR*) values indicating which *Z* values stand out as outliers in the *category-to-categoryall* integration.



Sample folder(s)	Lower level	Higher level	Reported vars	Output report	Column headers to eliminate	Merge with report	Add columns from relation table
*	peptide	protein	n	Npep2prot			
*	protein	proteinall	Z , FDR	Npep2prot_Quanprot	peptide	Npep2prot	protein2description
*	protein	category	n	Nprot2cat			
*	category	categoryall	Z , FDR	Nprot2cat_Quancat	protein	Nprot2cat	

Figure 39. The REPORT task table for workflow 1.

The following advanced options of the REPORT module have been used in this workflow (*Figure 39*):

- *Merge with report* designates the report file whose variables will be incorporated into the current report on the basis of their element identifiers. Thus, the *Npep2prot_Quanprot* report is first populated with the Z values of the *protein-to-proteinall* integration, and then added the number of peptides per protein coming from the *Npep2prot* report (*Figure 39*). This is possible because the *protein* level is present in both the *Npep2prot* and *Npep2prot_Quanprot* reports.
- *Column headers to eliminate* restricts the elements to be written to the *Output reports Npep2prot_Quanprot* and *Nprot2cat_Quancat* to those from the protein and category level, respectively. This prevents data duplication upon merging with the *Npep2prot* and *Nprot2cat* reports, as these contain also the peptide and protein level, respectively (*Figure 39*).
- *Add columns from relation table* appends a column with the description of the protein to the *Npep2prot_Quanprot* report. Again, this is possible because the protein level is present in both the *Npep2prot_Quanprot* report and the *protein2description* relation table (*Figure 39*).

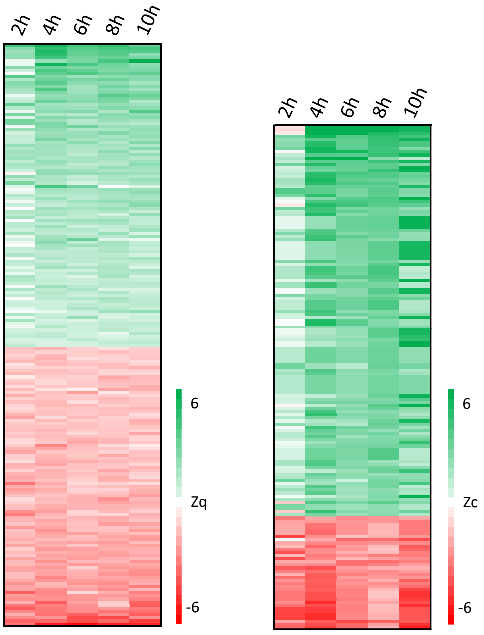


Figure 40. Relative abundance of proteins (Zq, Left) and functional categories (Zc, Right) retrieved from the Npep2prot_Quanprot and Nprot2cat_Quancat reports, respectively, generated with the REPORT module of Workflow1 (Figure 39).

As previously shown with SanXoT [2], iSanXoT workflow 1, based on the WSPP-SBT and the REPORT modules, reveals a slight tendency to increased protein abundance over time. In addition, the SBT module reveals that the categories found altered at 4, 8, and 10 h, but not those at 2 h, were essentially the same that were altered at 6 h (*Figure 40*).

Workflow execution

The workflow template and input files that are needed to execute this workflow can be downloaded from

https://github.com/CNIC-Proteomics/iSanXoT/wiki/studies/cases/templates/WSPP-SBT.zip

See the *Executing a workflow template* Section below for detailed instructions.

Workflow 2: Step-by-step quantification and sample combination in a labeled experiment

Experimental

We have applied workflow 2 to quantitative data obtained by González-Amor *et al*. [5] in their study of the contribution of interferon-stimulated gene 15 (ISG15) to vascular damage associated with hypertension. Protein extracts from mouse aortic tissue (16 samples from eight individuals:

two WT-Control, two ISG15-KO, two WT+AngII, and two ISG15-KO+AngII) were trypsin-digested, after which the resulting peptides were isobarically labelled using iTRAQ 8-plex and distributed in two labeled experiments that were analyzed by LC-MS/MS. Protein identification

was achieved by database searching against target and decoy sequences using Proteome Discoverer 2.1 (Thermo Fisher Scientific; see Section 3 below to learn how these data are adapted to be used with iSanXoT). The FDR for peptide identification was calculated using the refined method [7].

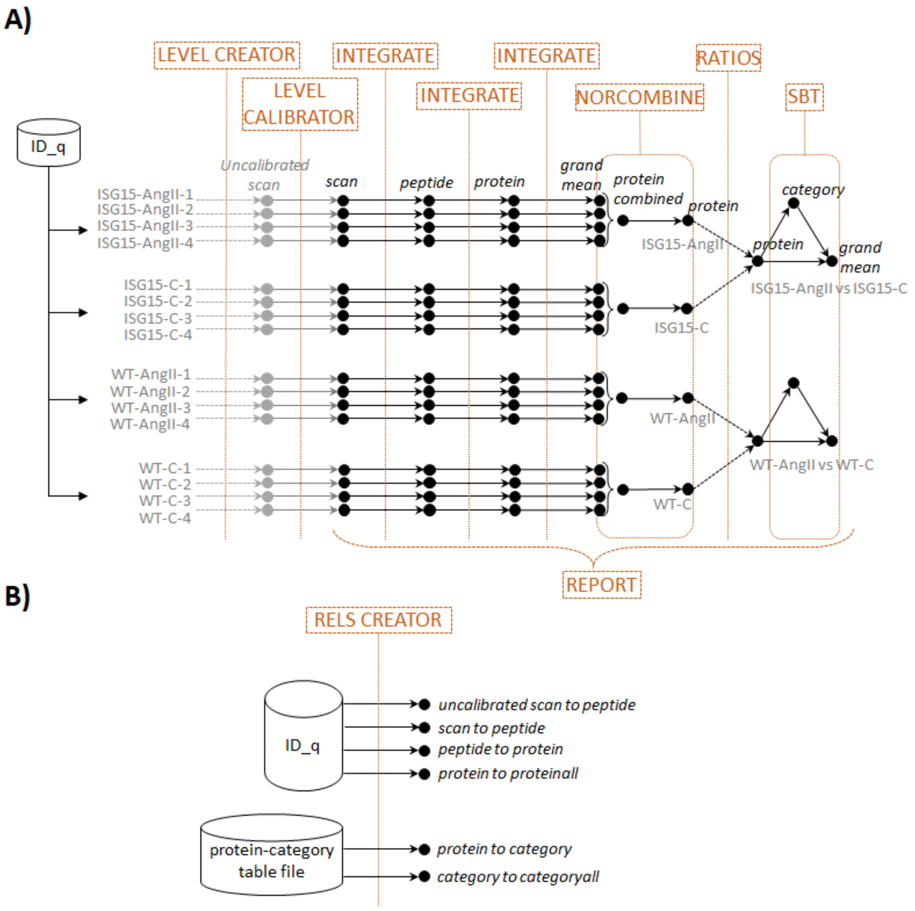
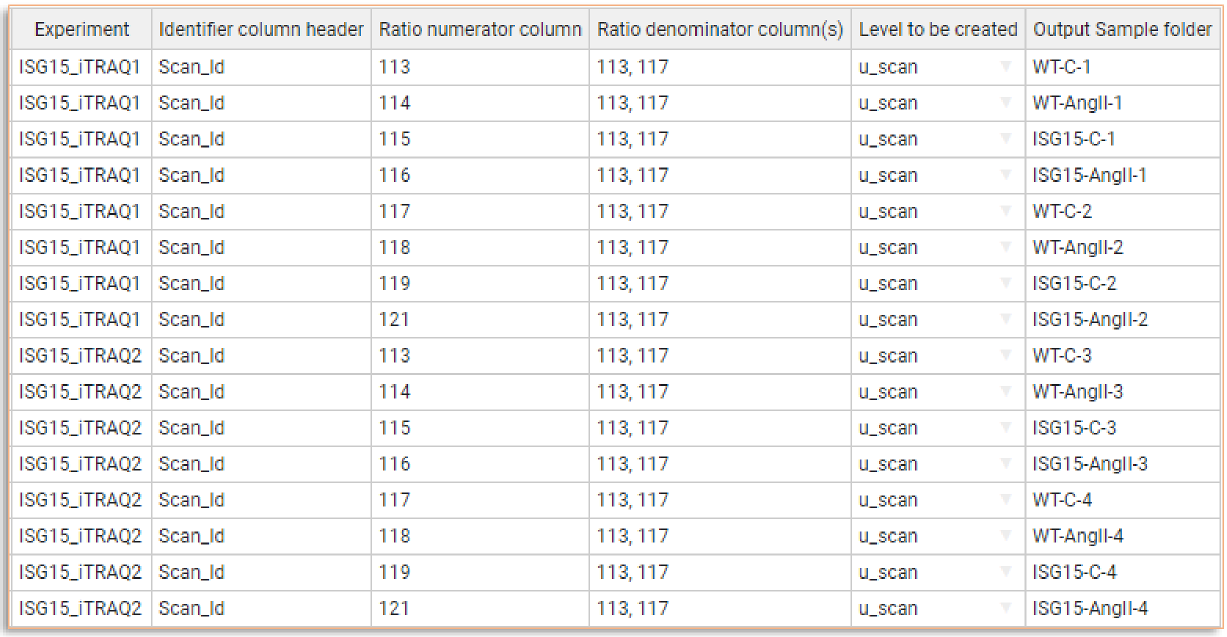


Figure 41. Scheme of workflow 2 (step-by-step quantification and sample combination in a labeled experiment) showing module components: LEVEL CREATOR, LEVEL CALIBRATOR, INTEGRATE, NORCOMBINE, RATIOS, SBT, and REPORT (A) and RELS CREATOR (B).

Workflow operation

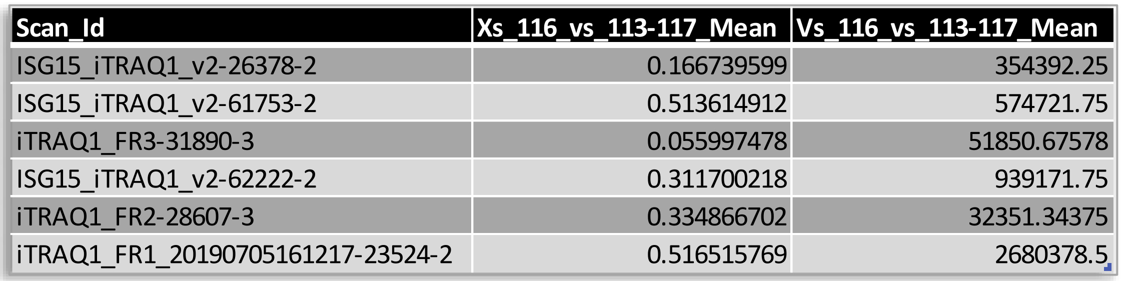
Workflow 2 comprises all six basic modules: LEVEL CREATOR, LEVEL CALIBRATOR, INTEGRATE, NORCOMBINE, RATIOS, and SBT plus the REPORT module (*Figure 41*). The starting module, LEVEL CREATOR, generates the level files, sample folders and log2 ratios indicated in the corresponding task table (*Figure 42*).



Experiment	Identifier column header	Ratio numerator column	Ratio denominator column(s)	Level to be created	Output Sample folder
ISG15_iTRAQ1	Scan_Id	113	113, 117	u_scan	WT-C-1
ISG15_iTRAQ1	Scan_Id	114	113, 117	u_scan	WT-AngII-1
ISG15_iTRAQ1	Scan_Id	115	113, 117	u_scan	ISG15-C-1
ISG15_iTRAQ1	Scan_Id	116	113, 117	u_scan	ISG15-AngII-1
ISG15_iTRAQ1	Scan_Id	117	113, 117	u_scan	WT-C-2
ISG15_iTRAQ1	Scan_Id	118	113, 117	u_scan	WT-AngII-2
ISG15_iTRAQ1	Scan_Id	119	113, 117	u_scan	ISG15-C-2
ISG15_iTRAQ1	Scan_Id	121	113, 117	u_scan	ISG15-AngII-2
ISG15_iTRAQ2	Scan_Id	113	113, 117	u_scan	WT-C-3
ISG15_iTRAQ2	Scan_Id	114	113, 117	u_scan	WT-AngII-3
ISG15_iTRAQ2	Scan_Id	115	113, 117	u_scan	ISG15-C-3
ISG15_iTRAQ2	Scan_Id	116	113, 117	u_scan	ISG15-AngII-3
ISG15_iTRAQ2	Scan_Id	117	113, 117	u_scan	WT-C-4
ISG15_iTRAQ2	Scan_Id	118	113, 117	u_scan	WT-AngII-4
ISG15_iTRAQ2	Scan_Id	119	113, 117	u_scan	ISG15-C-4
ISG15_iTRAQ2	Scan_Id	121	113, 117	u_scan	ISG15-AngII-4

Figure 42. The LEVEL CREATOR task table for workflow 2.

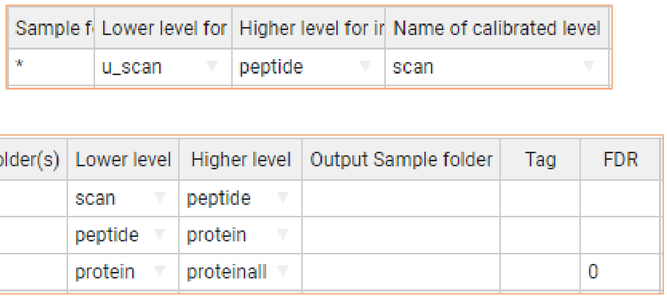
The output level files, *u_scan* (*uncalibrated scan*), display element identifiers together with their corresponding log2-ratio and statistical weight values (*Figure 43*). This *uncalibrated* weight is the inverse of the variance associated with the log2-ratio, and therefore a higher weight corresponds to a more accurate quantification.



Scan_Id	Xs_116_vs_113-117_Mean	Vs_116_vs_113-117_Mean
ISG15_iTRAQ1_v2-26378-2	0.166739599	354392.25
ISG15_iTRAQ1_v2-61753-2	0.513614912	574721.75
iTRAQ1_FR3-31890-3	0.055997478	51850.67578
ISG15_iTRAQ1_v2-62222-2	0.311700218	939171.75
iTRAQ1_FR2-28607-3	0.334866702	32351.34375
iTRAQ1_FR1_20190705161217-23524-2	0.516515769	2680378.5

Figure 43. Excerpt from one of the u_scan files generated by workflow 2 LEVEL CREATOR module showing element identifiers (left column), log2 ratios (center column), and statistical weights (right column).

Prior to their integration into a higher level, *u_scan* levels are calibrated to account for the variability of the LC-MS/MS analysis with the LEVEL CALIBRATOR module, which results in *scan* calibrated levels with recalculated statistical weights (*Figure 44*, Top). Then the INTEGRATE module performs the *scan-to-peptide*, *peptide-to-protein* and *protein-to-proteinall* integrations according to the module task table (*Figure 44*, Bottom). The protein outliers are not discarded in this integration as prompted by the 0 value indicated for the FDR parameter.

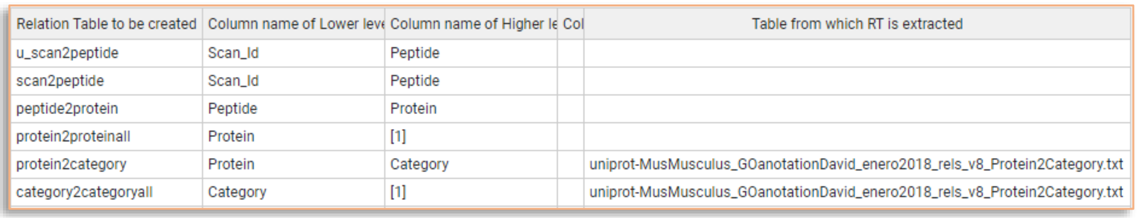


Sample f	Lower level for	Higher level for ir	Name of calibrated level
*	u_scan	peptide	scan

Sample folder(s)	Lower level	Higher level	Output Sample folder	Tag	FDR
*	scan	peptide			
*	peptide	protein			
*	protein	proteinall			0

Figure 44 . The LEVEL CALIBRATOR (Top) and INTEGRATE (Bottom) task tables for workflow 2.

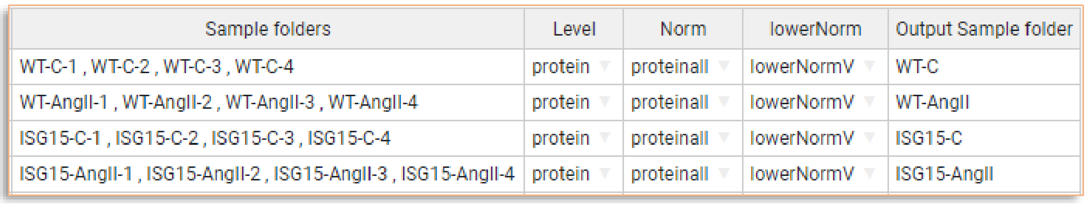
Every integration requires a specific relation table linking the corresponding lower- and higher-level elements that is generated by RELS CREATOR (*Figure 41*B and *Figure 45*). Thus, the *peptide2protein* relation table (*Figure 45*) describes the relationships between the peptide level (*i.e.* the lower level) and the protein level (*i.e.* the higher level). Note that the elements of the *protein2category* relation table were retrieved from a text file containing functional annotations from the Gene Ontology knowledgebase [5].



Relation Table to be created	Column name of Lower leve	Column name of Higher le	Col	Table from which RT is extracted
u_scan2peptide	Scan_Id	Peptide		
scan2peptide	Scan_Id	Peptide		
peptide2protein	Peptide	Protein		
protein2proteinall	Protein	[1]		
protein2category	Protein	Category		uniprot-MusMusculus_GOanotationDavid_enero2018_rels_v8_Protein2Category.txt
category2categoryall	Category	[1]		uniprot-MusMusculus_GOanotationDavid_enero2018_rels_v8_Protein2Category.txt

Figure 45. The RELS CREATOR task table for workflow 2.

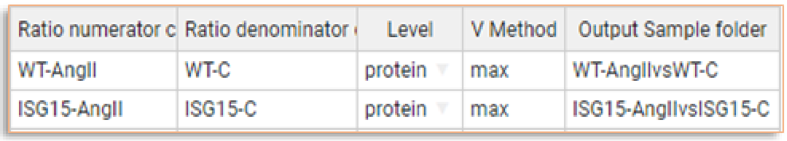
Next, workflow 2 resorts to the NORCOMBINE basic module (*Figure 41*A) to combine biological replicates, which results in the four new samples indicated on *Figure 46*: WT-C, WT-AngII, ISG15-C, and ISG15-AngII. The module task table indicates that samples were combined at the *protein* level using the *proteinall* level for normalization. NORCOMBINE also requires the user to specify which *lowerNorm* files contain the necessary data for the samples to be combined. These *lowerNorm* files, previously generated by the INTEGRATE module, contain lower-level identifiers along with the corresponding centered log2-ratio values and either the integration statistical weight (in the case of *lowerNormV*, as chosen in this workflow) or the variance (for *lowerNormW*) (*Figure 46*). NORCOMBINE first generates merged data files and relation tables that are later integrated to the above-described four grouped levels.



Sample folders	Level	Norm	lowerNorm	Output Sample folder
WT-C-1 , WT-C-2 , WT-C-3 , WT-C-4	protein	proteinall	lowerNormV	WT-C
WT-AngII-1 , WT-AngII-2 , WT-AngII-3 , WT-AngII-4	protein	proteinall	lowerNormV	WT-AngII
ISG15-C-1 , ISG15-C-2 , ISG15-C-3 , ISG15-C-4	protein	proteinall	lowerNormV	ISG15-C
ISG15-AngII-1 , ISG15-AngII-2 , ISG15-AngII-3 , ISG15-AngII-4	protein	proteinall	lowerNormV	ISG15-AngII

Figure 46. The NORCOMBINE task table for workflow 2.

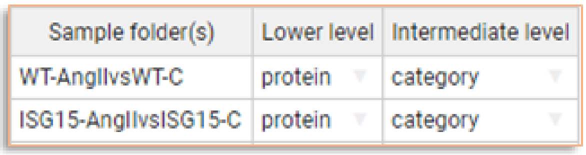
The four new samples (grouped levels or sample groups) are then used by the RATIOS basic module to calculate two ratios: *WT-AngIIvsWT-C*, where wild-type AngII-treated animals are compared to controls; and *ISG15-AngIIvsISG15-C*, where ISG15 AngII-treated animals are compared to controls (*Figure 47*). The module calculates the new log2 ratios as the difference between *Ratio numerator* and *Ratio denominator* values, whereas the corresponding statistical weight is assessed according to the *V Method* indicated in the task table (*max* in this case, which means the maximum weight between the samples indicated under *Ratio numerator* and *Ratio denominator*. The other two choices are *avg*, the average of the two weights; and *form*, $1/(1/V_n + 1/V_d)$, where V_n and V_d are the statistical weight of the samples indicated under *Ratio numerator* and *Ratio denominator*, respectively).



Ratio numerator c	Ratio denominator	Level	V Method	Output Sample folder
WT-AngII	WT-C	protein	max	WT-AngIIvsWT-C
ISG15-AngII	ISG15-C	protein	max	ISG15-AngIIvsISG15-C

Figure 47. The RATIOS task table for workflow 2.

The last basic module executed in workflow 2, SBT (*Figure 48*), relies on the systems biology triangle algorithm [3] to perform an integration between the *Lower level* (*protein* in this case) and the corresponding grand mean using the variance previously obtained in the integration between the *Lower level* and an *Intermediate level* (*category* in this case, *Figure 48*). The way it is used here, the SBT module reveals functional category changes originated by the coordinated behaviour of their protein components. The integrations involved, *protein-to-category*, *protein-to-proteinall*, and *category-to-categoryall*, require the relation tables previously generated with RELS CREATOR (*Figure 45*). As mentioned above, the elements of the *protein2category* relation table were retrieved from a text file containing functional annotations from the Gene Ontology knowledgebase [3].

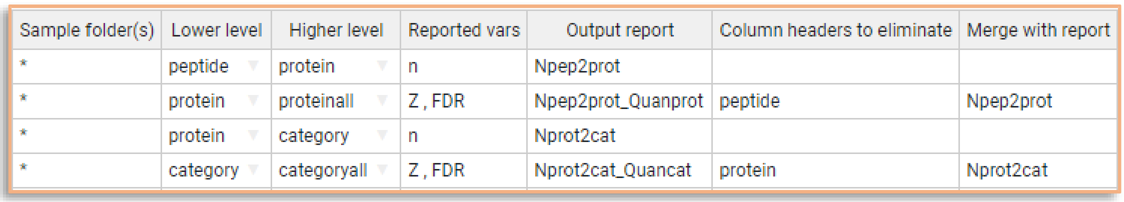


Sample folder(s)	Lower level	Intermediate level
WT-AngIIvsWT-C	protein	category
ISG15-AngIIvsISG15-C	protein	category

Figure 48. The SBT task table for workflow 2.

Finally, the REPORT module (*Figure 49*) is used in this workflow to collect the following statistical variables for all samples, namely the 16 individual samples generated with LEVEL CREATOR (*e.g. Ind/WT-C-1* and *Ind/ISG15-AngII-1*, *Figure 42*), the four combined samples obtained with NORCOMBINE (*WT-C, WT-AngII, ISG15-C*, and *ISG15-AngII*, *Figure 46*), and the two ratio samples created by RATIOS (*WT-AngIIvsWT-C* and ISG15-AngIIvsISG15-C, *Figure 47*), as prompted by the asterisk:

- Number of peptides per protein (*n*), obtained from the *peptide-to-protein* integration;
- Mean-corrected protein log2 ratios expressed in units of standard deviation (*Z*), calculated from the *protein-to-proteinall* integration;
- False Discovery Rate (*FDR*) values indicating which *Z* values stand out as outliers in the *protein-to-proteinall* integration;
- Number of proteins per category (*n*), obtained from the *protein-to-category integration*;
- Mean-corrected category log2 ratios expressed in units of standard deviation (*Z*), calculated from the *category-to-categoryall* integration;
- False Discovery Rate (*FDR*) values indicating which *Z* values stand out as outliers in the *category-to-categoryall* integration.



Sample folder(s)	Lower level	Higher level	Reported vars	Output report	Column headers to eliminate	Merge with report
*	peptide	protein	n	Npep2prot		
*	protein	proteinall	Z , FDR	Npep2prot_Quanprot	peptide	Npep2prot
*	protein	category	n	Nprot2cat		
*	category	categoryall	Z , FDR	Nprot2cat_Quancat	protein	Nprot2cat

Figure 49. The REPORT task table for workflow 2.

The following advanced options of the REPORT module have been used in this workflow (*Figure 49*):

- *Merge with report* designates the report file whose variables will be incorporated into the current report on the basis of their element identifiers. Thus, for example, the *Npep2prot_Quanprot* report is first populated with the *Z* values of the *protein-to-proteinall* integration, and then added the number of peptides per protein coming from the *Npep2prot* report (*Figure 49*). This is possible because the *protein* level is present in both report files.
- *Column headers to eliminate* restricts the elements to be written to the *Output reports Npep2prot_Quanprot* and *Nprot2cat_Quancat* to those from the protein and category level, respectively. This prevents data duplication upon merging with the *Npep2prot* and *Nprot2cat* reports, as these contain also the peptide and protein level, respectively (*Figure 49*).

As previously shown with SanXoT [5], iSanXoT analysis revealed a coordinated alteration of proteins implicated in cardiovascular function, extracellular matrix and remodeling, and vascular redox state in aorta from AngII-infused ISG15-KO mice (*Figure 50*).

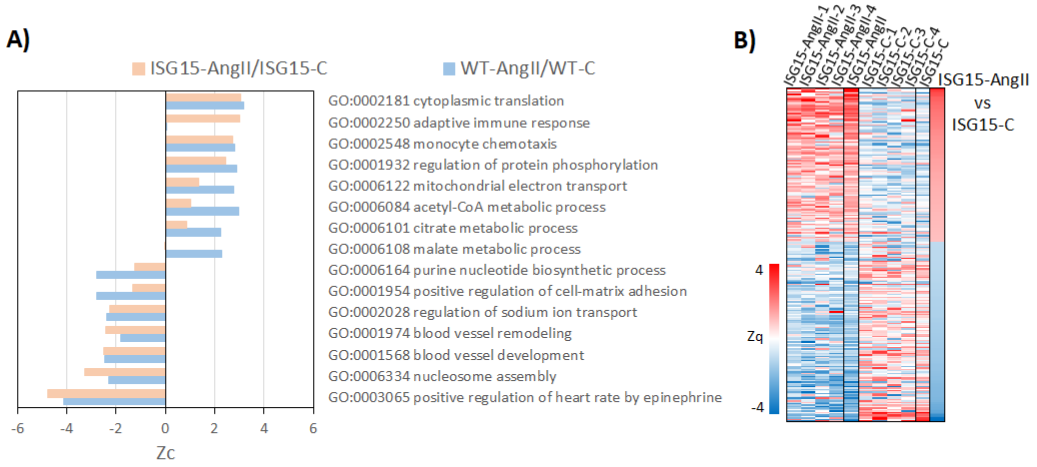


Figure 50. Differential abundance of functional categories (A) and proteins (B) revealed by workflow 2 based on the mean-corrected log2-ratio expressed in units of standard deviation at the category (Zc) and protein (Zq) level, respectively.

Workflow execution

The workflow template and input files that are needed to execute this workflow can be downloaded from

https://github.com/CNIC-Proteomics/iSanXoT/wiki/studies/cases/templates/WSPP_NORCOM_RATIOS_SBT.zip

See the *Executing a workflow template* Section below for detailed instructions.

Workflow 3: Quantification of posttranslationally modified peptides in a labeled experiment

Experimental

This workflow was used to quantitate Cys-oxidized peptides in mouse embryonic fibroblast (MEF) preparations subjected to chemical oxidation with diamide, an experiment that served to illustrate the comparative performance of on-filter (FASILOX) and in-gel (GELSILOX) approaches to study the thiol redox proteome [4]. MEF samples were incubated with diamide (treated group) or PBS (control group) for 10 min (four biological replicates per condition). The FASP filter was used to carry out the sequential alkylation and reduction steps needed to separately label reduced and oxidized Cys residues, after which the proteins were digested with trypsin and the resulting peptides were isobaricaly labeled with iTRAQ 8-plex and analyzed by LC-MS/MS. Protein identification was achieved with Proteome Discoverer 1.4 (Thermo Fisher Scientific; see Section 3 below to learn how these data are adapted to be used with iSanXoT). The FDR for peptide identification was calculated using the refined method [7].

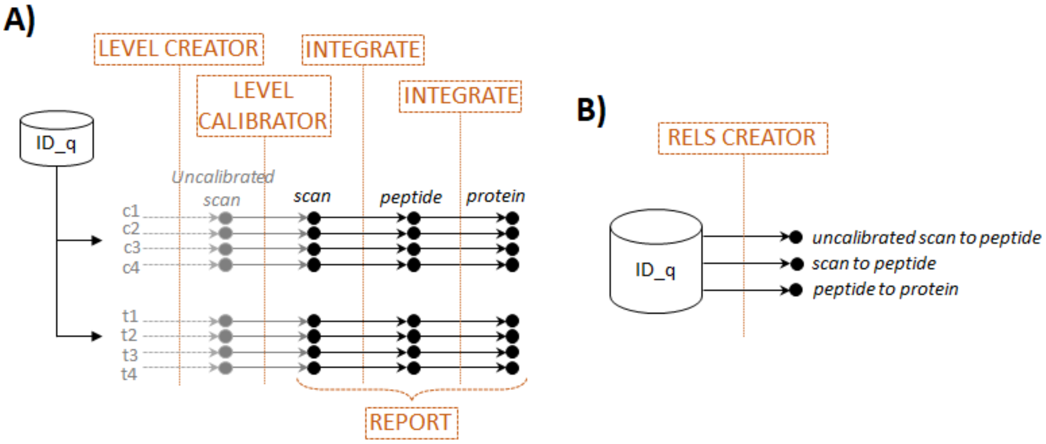
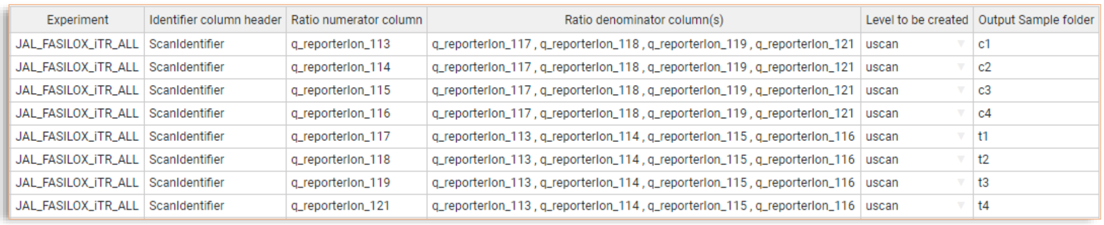


Figure 51. Scheme of workflow 3 (quantification of posttranslationally modified peptides in a labeled experiment) showing module components: LEVEL CREATOR, LEVEL CALIBRATOR, INTEGRATE, and REPORT (A) and RELS CREATOR (B).

Workflow operation

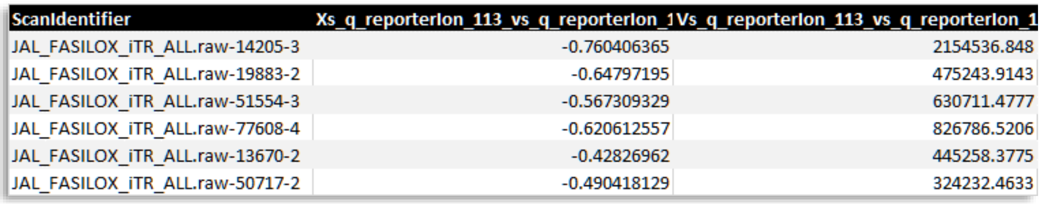
Workflow 3 comprises the basic modules LEVEL CREATOR, LEVEL CALIBRATOR, and INTEGRATE, as well as the RELS CREATOR and REPORT modules (*Figure 51*). The starting module, LEVEL CREATOR, generates the level files, sample folders and log2 ratios indicated in the corresponding task table (*Figure 52*).



Experiment	Identifier column header	Ratio numerator column	Ratio denominator column(s)	Level to be created	Output Sample folder
JAL_FASILOX_iTR_ALL	ScanIdentifier	q_reporterIon_113	q_reporterIon_117 , q_reporterIon_118 , q_reporterIon_119 , q_reporterIon_121	uscan	c1
JAL_FASILOX_iTR_ALL	ScanIdentifier	q_reporterIon_114	q_reporterIon_117 , q_reporterIon_118 , q_reporterIon_119 , q_reporterIon_121	uscan	c2
JAL_FASILOX_iTR_ALL	ScanIdentifier	q_reporterIon_115	q_reporterIon_117 , q_reporterIon_118 , q_reporterIon_119 , q_reporterIon_121	uscan	c3
JAL_FASILOX_iTR_ALL	ScanIdentifier	q_reporterIon_116	q_reporterIon_117 , q_reporterIon_118 , q_reporterIon_119 , q_reporterIon_121	uscan	c4
JAL_FASILOX_iTR_ALL	ScanIdentifier	q_reporterIon_117	q_reporterIon_113 , q_reporterIon_114 , q_reporterIon_115 , q_reporterIon_116	uscan	t1
JAL_FASILOX_iTR_ALL	ScanIdentifier	q_reporterIon_118	q_reporterIon_113 , q_reporterIon_114 , q_reporterIon_115 , q_reporterIon_116	uscan	t2
JAL_FASILOX_iTR_ALL	ScanIdentifier	q_reporterIon_119	q_reporterIon_113 , q_reporterIon_114 , q_reporterIon_115 , q_reporterIon_116	uscan	t3
JAL_FASILOX_iTR_ALL	ScanIdentifier	q_reporterIon_121	q_reporterIon_113 , q_reporterIon_114 , q_reporterIon_115 , q_reporterIon_116	uscan	t4

Figure 52. The LEVEL CREATOR task table for workflow 3.

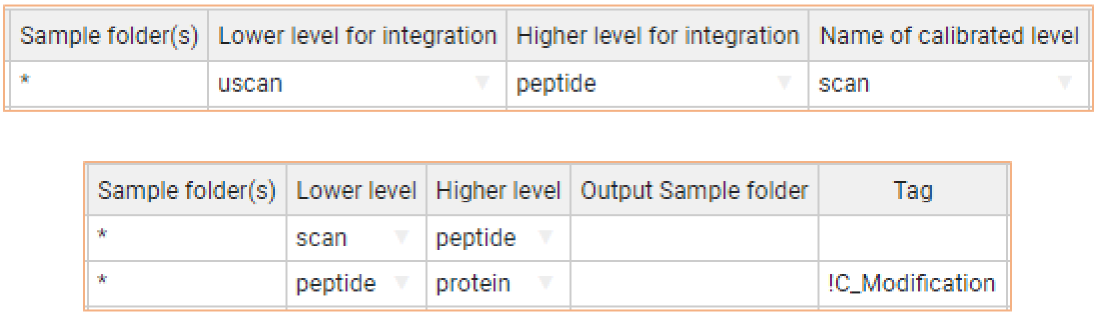
The output level files, *uscan* (*uncalibrated scan*), display element identifiers together with their corresponding log2-ratio and statistical weight values (*Figure 53*). This uncalibrated weight is the inverse of the variance associated with the log2-ratio, and therefore a higher weight corresponds to a more accurate quantification.



ScanIdentifier	Xs_q_reporterIon_113_vs_q_reporterIon_1	Vs_q_reporterIon_113_vs_q_reporterIon_1
JAL_FASILOX_iTR_ALL.raw-14205-3	-0.760406365	2154536.848
JAL_FASILOX_iTR_ALL.raw-19883-2	-0.64797195	475243.9143
JAL_FASILOX_iTR_ALL.raw-51554-3	-0.567309329	630711.4777
JAL_FASILOX_iTR_ALL.raw-77608-4	-0.620612557	826786.5206
JAL_FASILOX_iTR_ALL.raw-13670-2	-0.42826962	445258.3775
JAL_FASILOX_iTR_ALL.raw-50717-2	-0.490418129	324232.4633

Figure 53. Excerpt from one of the uscan files generated by workflow 3 LEVEL CREATOR module showing element identifiers (left column), log2 ratios (center column) and statistical weights (right column).

Prior to their integration into a higher level, *uscan* levels are calibrated to account for the variability of the LC-MS/MS analysis with the LEVEL CALIBRATOR module, which results in scan calibrated levels with recalculated statistical weights (*Figure 54*, Top). Then the INTEGRATE module performs the *scan-to-peptide* and *peptide-to- protein* integrations according to the module task table (*Figure 54*, Bottom). The latter integration will provide the differential abundance of peptides (including posttranslationally modified peptides) regardless of the differential abundance of proteins, and the authors chose not to include the heterogeneous population of Cys-containing peptides for the calculation of variance in the *peptide-to-protein* integration [4].

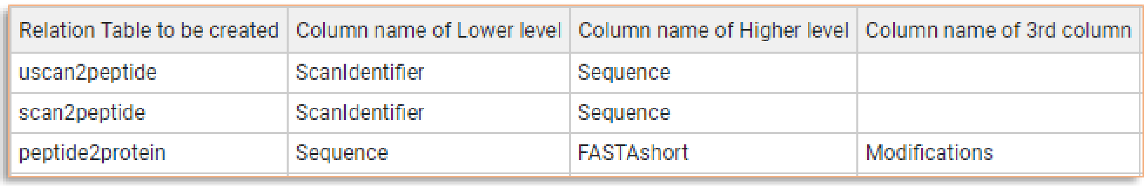


Sample folder(s)	Lower level for integration	Higher level for integration	Name of calibrated level
*	uscan	peptide	scan

Sample folder(s)	Lower level	Higher level	Output Sample folder	Tag
*	scan	peptide		
*	peptide	protein		!C_Modification

Figure 54. The LEVEL CALIBRATOR (Top) and INTEGRATE (Bottom) task tables for workflow 3.

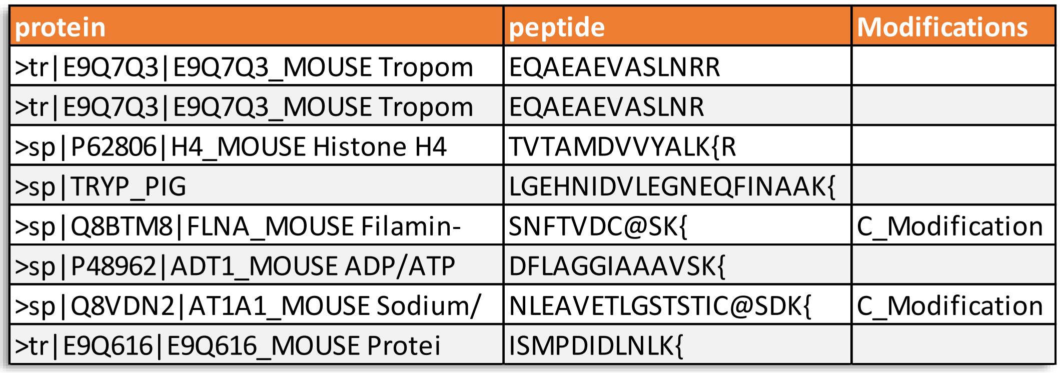
Thus, the *Tag* advanced option of the INTEGRATE module task table displays *!C_Modification*, which means that the *peptide* (lower level) elements marked with this tag in the *peptide2protein* relation table (*Figure 55*) will not be considered in the calculation of the variance in the *peptide-to-protein* integration. Every integration requires a specific relation table linking the corresponding lower- and higher-level elements that is generated by RELS CREATOR (*Figure 51*B and *Figure 55*).



Relation Table to be created	Column name of Lower level	Column name of Higher level	Column name of 3rd column
uscan2peptide	ScanIdentifier	Sequence	
scan2peptide	ScanIdentifier	Sequence	
peptide2protein	Sequence	FASTAshort	Modifications

Figure 55. The RELS CREATOR task table for workflow 3.

In particular, the *peptide2protein* relation table describes the relationships between the peptide level (*i.e*. the lower level) and the protein level (*i.e*. the higher level), and in this case includes a third column headed by *Modifications* where Cys-containing peptides have been tagged with *C_Modification* (*Figure 56*). Accordingly, Cys-containing peptides will not be considered for the calculation of variance in the *peptide-to-protein* integration.



protein	peptide	Modifications
>tr|E9Q7Q3|E9Q7Q3_MOUSE Tropom	EQAEAEVASLNRR	
>tr|E9Q7Q3|E9Q7Q3_MOUSE Tropom	EQAEAEVASLNR	
>sp|P62806|H4_MOUSE Histone H4	TVTAMDVVYALK{R	
>sp|TRYP_PIG	LGEHNIDVLEGNEQFINAAK{	
>sp|Q8BTM8|FLNA_MOUSE Filamin-	SNFTVDC@SK{	C_Modification
>sp|P48962|ADT1_MOUSE ADP/ATP	DFLAGGIAAAVSK{	
>sp|Q8VDN2|AT1A1_MOUSE Sodium/	NLEAVETLGSTSTIC@SDK{	C_Modification
>tr|E9Q616|E9Q616_MOUSE Protei	ISMPDIDLNLK{	

Figure 56. Excerpt from the peptide2protein relation table that links peptides to proteins. Note the use of a third column to tag Cys-containing peptides, which will be later excluded from the calculation of variance in the peptide-to-protein integration.

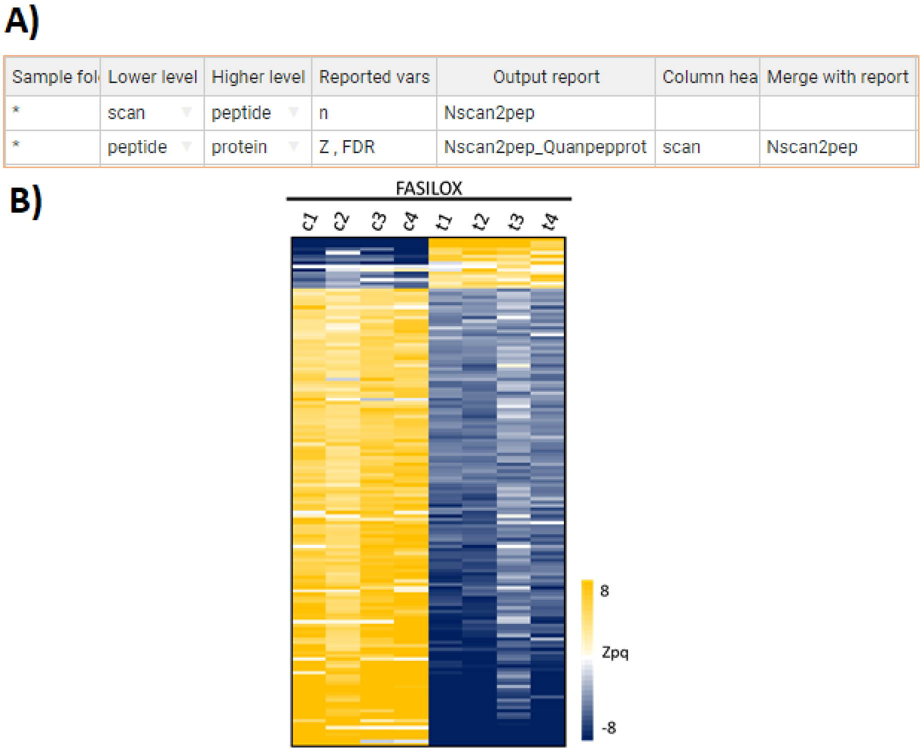
Finally, the REPORT module is used ion this workflow to collect the statistical variables generated by the *peptide-to-protein* integration for all eight samples (*c1, c2, c3, c4, t1, t2, t3,* and *t4*), as prompted by the asterisk (*Figure 57*A):

- Number of scans per peptide (*n*), obtained from the *scan-to-peptide* integration;
- Peptide log2 ratios expressed in units of standard deviation corrected by the protein mean (*Z*), calculated from the *peptide-to-protein* integration;
- False Discovery Rate (*FDR*) values indicating which *Z* values stand out as outliers in the integration.

The advanced options of the REPORT module (*Figure 57*A):

- *Merge with report* designates the merge of the number of scans per peptide from the *Nscan2pep* report and the Z values and FDR of the *peptide-to-protein* that will be saved in the *Nscan2pep_Quanpepprot report*.
- *Column headers columns to eliminate* restricts the elements to be written to the Output reports *Nscan2pep_Quanpepprot* to those from the scan level.

Figure 57B shows the relative abundance of Cys-containing peptides provided by the *peptide-to-protein* integration.



Sample fol	Lower level	Higher level	Reported vars	Output report	Column hea	Merge with report
*	scan	peptide	n	Nscan2pep		
*	peptide	protein	Z , FDR	Nscan2pep_Quanpepprot	scan	Nscan2pep

Figure 57. (A) The REPORT module task table for workflow 3. (B) Relative abundance of Cys-containing peptides in MEF samples measured by peptide log2 ratios expressed in units of standard deviation corrected by the protein mean (Zpq).

Workflow execution

The workflow template and input files that are needed to execute this workflow can be downloaded from

https://github.com/CNIC-Proteomics/iSanXoT/wiki/studies/cases/templates/WSPP_PTM.zip

See the *Executing a workflow template* Section below for detailed instructions.

Workflow 4: Label-free quantification

Experimental

This workflow has been applied to quantitative data obtained in a multicenter study aimed at evaluating different bioinformatics tools [6]. The authors generated two hybrid proteome samples consisting of tryptic digests of human, yeast and *Escherichia coli* proteins mixed in two different defined proportions (13:3:4 for *a* samples and 13:6:1 for b samples). Quadruplicate peptide samples were analyzed by LC-MS/MS, after which peptide identification and quantification were carried out with MaxQuant [8], among other software packages (see Section below to learn how these data are adapted to be used with iSanXoT).

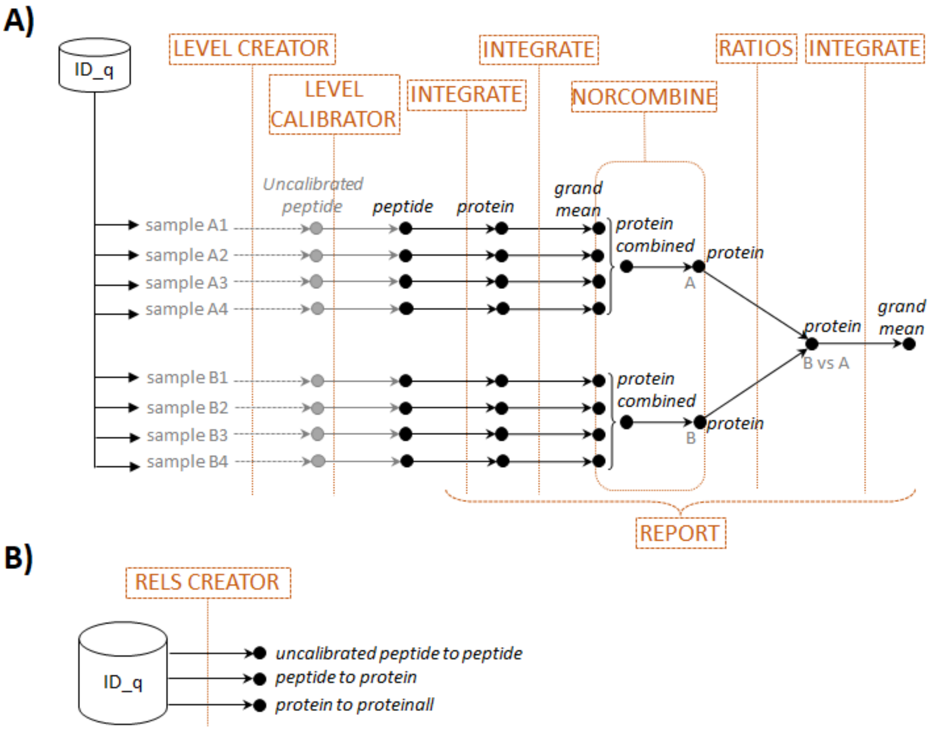
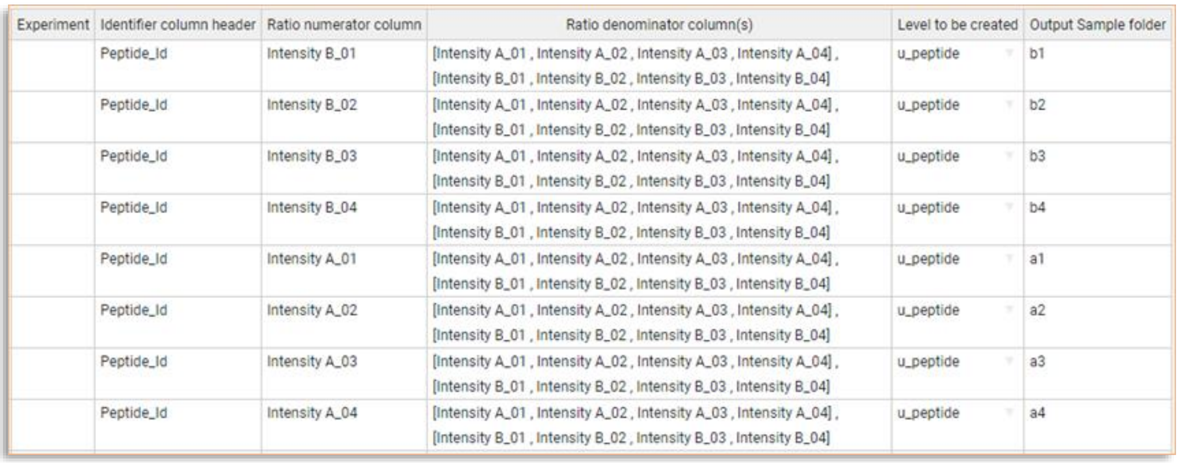


Figure 58. Scheme of workflow 4 (label-free quantification) showing module components: LEVEL CREATOR, LEVEL CALIBRATOR, INTEGRATE, NORCOMBINE, and REPORT (A) and RELS CREATOR (B).

Workflow operation

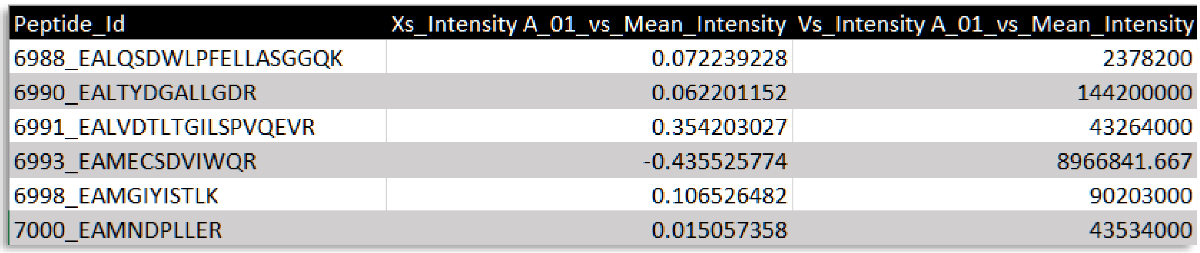
Workflow 4 includes the basic iSanXoT modules LEVEL CREATOR, LEVEL CALIBRATOR, INTEGRATE, NORCOMBINE, and RATIOS along with the REPORT and RELS CREATOR modules (*Figure 58*). The starting module, LEVEL CREATOR, generates the level files, sample folders and log2 ratios indicated in the corresponding task table (*Figure 59*) based on the quantitative data at the peptide level obtained with MaxQuant for replicate *a* and *b* samples. To calculate the log2-ratio denominator, first the averages of the four *a* and the four *b* samples are calculated separately (as indicated by the square brackets), then the average of the two average values is calculated (as indicated by the comma).



Experiment	Identifier column header	Ratio numerator column	Ratio denominator column(s)	Level to be created	Output Sample folder
	Peptide_Id	Intensity B_01	[Intensity A_01 , Intensity A_02 , Intensity A_03 , Intensity A_04] , [Intensity B_01 , Intensity B_02 , Intensity B_03 , Intensity B_04]	u_peptide	b1
	Peptide_Id	Intensity B_02	[Intensity A_01 , Intensity A_02 , Intensity A_03 , Intensity A_04] , [Intensity B_01 , Intensity B_02 , Intensity B_03 , Intensity B_04]	u_peptide	b2
	Peptide_Id	Intensity B_03	[Intensity A_01 , Intensity A_02 , Intensity A_03 , Intensity A_04] , [Intensity B_01 , Intensity B_02 , Intensity B_03 , Intensity B_04]	u_peptide	b3
	Peptide_Id	Intensity B_04	[Intensity A_01 , Intensity A_02 , Intensity A_03 , Intensity A_04] , [Intensity B_01 , Intensity B_02 , Intensity B_03 , Intensity B_04]	u_peptide	b4
	Peptide_Id	Intensity A_01	[Intensity A_01 , Intensity A_02 , Intensity A_03 , Intensity A_04] , [Intensity B_01 , Intensity B_02 , Intensity B_03 , Intensity B_04]	u_peptide	a1
	Peptide_Id	Intensity A_02	[Intensity A_01 , Intensity A_02 , Intensity A_03 , Intensity A_04] , [Intensity B_01 , Intensity B_02 , Intensity B_03 , Intensity B_04]	u_peptide	a2
	Peptide_Id	Intensity A_03	[Intensity A_01 , Intensity A_02 , Intensity A_03 , Intensity A_04] , [Intensity B_01 , Intensity B_02 , Intensity B_03 , Intensity B_04]	u_peptide	a3
	Peptide_Id	Intensity A_04	[Intensity A_01 , Intensity A_02 , Intensity A_03 , Intensity A_04] , [Intensity B_01 , Intensity B_02 , Intensity B_03 , Intensity B_04]	u_peptide	a4

Figure 59. The LEVEL CREATOR task table for workflow 4.

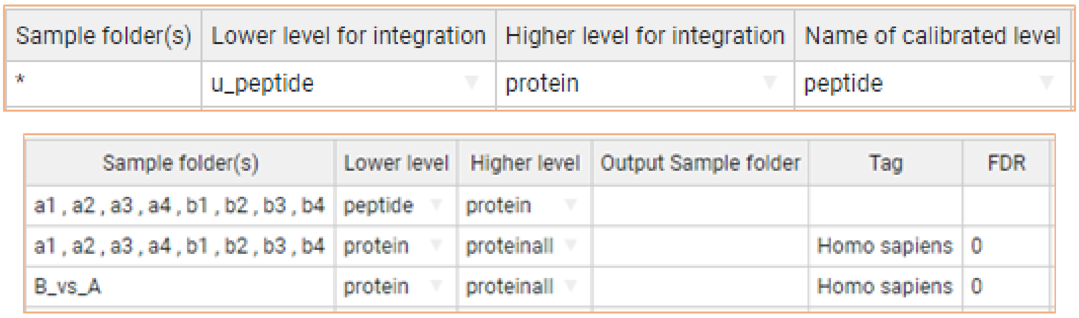
This ensures that no log2-ratio is calculated when all four values are missing in either the *a* or the *b* sample group. The eight *u_peptide* (uncalibrated peptide) output level files display element identifiers together with their corresponding log2-ratio and statistical weight values (*Figure 60*).



Peptide_Id	Xs_Intensity A_01_vs_Mean_Intensity	Vs_Intensity A_01_vs_Mean_Intensity
6988_EALQSDWLPFELLASGGQK	0.072239228	2378200
6990_EALTYDGALLGDR	0.062201152	144200000
6991_EALVDTLTGILSPVQEVR	0.354203027	43264000
6993_EAMECSDVIWQR	-0.435525774	8966841.667
6998_EAMGIYISTLK	0.106526482	90203000
7000_EAMNDPLLER	0.015057358	43534000

Figure 60. Excerpt from one of the u_peptide files generated by workflow 4 LEVEL CREATOR module showing element identifiers (left column), log2 ratios (center column) and statistical weights (right column).

This uncalibrated weight is the inverse of the variance associated with the log2-ratio, and therefore a higher weight corresponds to a more accurate quantification. Prior to their integration into a higher level, *u_peptide* levels are calibrated with the LEVEL CALIBRATOR module, which results in *peptide* calibrated levels with recalculated statistical weights (*Figure 61*, Top). Then the INTEGRATE module performs the *peptide-to-protein* and *protein-to-proteinall* integrations according to the module task table (*Figure 61*, Bottom).

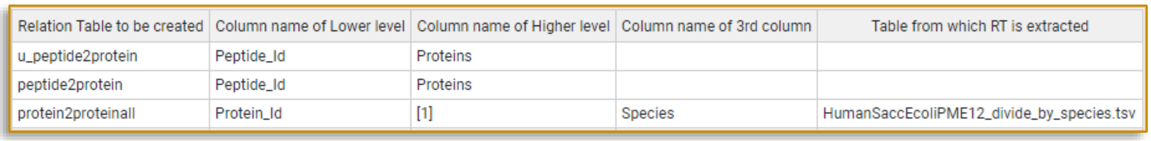


Sample folder(s)	Lower level for integration	Higher level for integration	Name of calibrated level
*	u_peptide	protein	peptide

Sample folder(s)	Lower level	Higher level	Output Sample folder	Tag	FDR
a1 , a2 , a3 , a4 , b1 , b2 , b3 , b4	peptide	protein			
a1 , a2 , a3 , a4 , b1 , b2 , b3 , b4	protein	proteinall		Homo sapiens	0
B_vs_A	protein	proteinall		Homo sapiens	0

Figure 61. The LEVEL CALIBRATOR (Top) and INTEGRATE (Bottom) task tables for workflow 4.

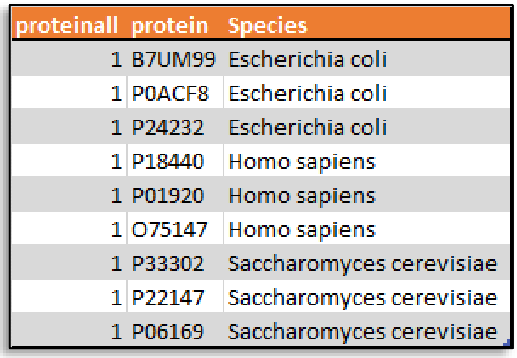
Every integration requires a specific relation table linking the corresponding lower- and higher-level elements that is generated by RELS CREATOR (*Figure 58*B and *Figure 62*); for instance, the *peptide2protein* relation table mentioned describes the relationships between the peptide level (*i.e.* the lower level) and the protein level (*i.e.* the higher level).



Relation Table to be created	Column name of Lower level	Column name of Higher level	Column name of 3rd column	Table from which RT is extracted
u_peptide2protein	Peptide_Id	Proteins		
peptide2protein	Peptide_Id	Proteins		
protein2proteinall	Protein_Id	[1]	Species	HumanSaccEcoliPME12_divide_by_species.tsv

Figure 62. The RELS CREATOR task table for workflow 4.

As mentioned above, replicate *a* and *b* samples contain different proportions of yeast and *Escherichia coli* peptides (3:4 and 6:1, respectively) and the same proportion of human peptides; for this reason, only human proteins were considered in the calculation of the variance for the *protein-to-proteinall* integrations, as indicated by the *Tag* parameter used in the INTEGRATE task table (*Figure 61*, Bottom; note that no outliers were removed in this integration as prompted by the *0* value indicated for the *FDR* parameter). Accordingly, the *protein2proteinall* relation table required by this integration includes a third column headed by *Species* where proteins have been tagged with their corresponding species (*Figure 63*). This relation table was built based on the file *HumanSaccEcoliPME12_..._species.tsv*.

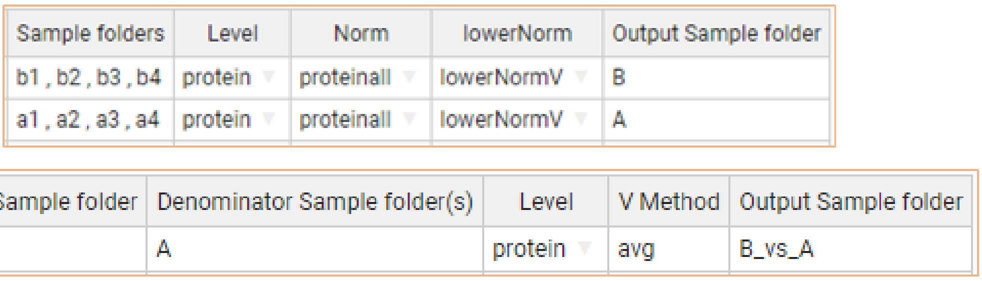


proteinall	protein	Species
1	B7UM99	Escherichia coli
1	P0ACF8	Escherichia coli
1	P24232	Escherichia coli
1	P18440	Homo sapiens
1	P01920	Homo sapiens
1	O75147	Homo sapiens
1	P33302	Saccharomyces cerevisiae
1	P22147	Saccharomyces cerevisiae
1	P06169	Saccharomyces cerevisiae

Figure 63. Excerpt from the protein2proteinall workflow 4 relation table that links proteins to a constant value representing the protein grand mean. Note the use of a third column to tag proteins with their corresponding species for later species-specific protein-to-proteinall integration.

Next, the four individual replicate *a* (*a1, a2, a3,* and *a4*) and *b* (*b1, b2, b3,* and *b4*) samples initially generated with LEVEL CREATOR (*Figure 59*) are combined into samples *A* and *B*, respectively, using the NORCOMBINE basic module (*Figure 64*, Top). The module task table indicates that samples were combined at the *protein* level using the *proteinall* level for normalization. NORCOMBINE also requires the user to specify which *lowerNorm* files contain the necessary data for the samples to be combined. These *lowerNorm* files, previously generated by the INTEGRATE module (*Figure 61*, Bottom), contain lower-level identifiers along with the corresponding centered log2-ratio values and either the integration statistical weight (in the case of *lowerNormV*, as chosen in this workflow) or the variance (for *lowerNormW*) (*Figure 64*, Top). NORCOMBINE first generates merged data files and relation tables that are later integrated to the above-described combined samples *A* and *B*.

To compared these two samples, first new log-2 ratios and statistical weights are calculated using the RATIOS basic module (*Figure 64*, Bottom), after which a *protein-to-proteinall* integration is carried out for the newly-generated *B_vs_A* sample (*Figure 61*, Bottom).



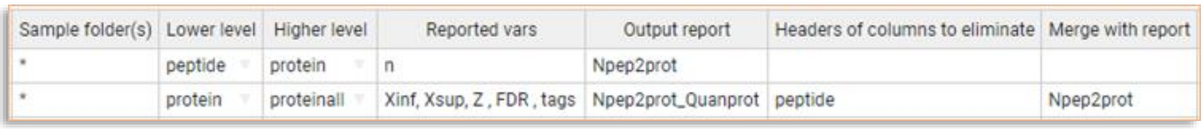
Sample folders	Level	Norm	lowerNorm	Output Sample folder
b1 , b2 , b3 , b4	protein	proteinall	lowerNormV	B
a1 , a2 , a3 , a4	protein	proteinall	lowerNormV	A

Numerator Sample folder	Denominator Sample folder(s)	Level	V Method	Output Sample folder
B	A	protein	avg	B_vs_A

Figure 64. The NORCOMBINE (Top) and RATIOS (Bottom) task tables for workflow 4.

Finally, the REPORT module (*Figure 65*) is used in this workflow to collect variables for all the samples involved (as prompted by the asterisk), namely the eight individual samples generated with LEVEL CREATOR (*a1, a2, a3, a4, b1, b2, b3,* and *b4*; *Figure 59*), the combined samples *A* and *B* obtained with NORCOMBINE (*Figure 64*, Top), and the *B_vs_A* sample generated by RATIOS (*Figure 64*, Bottom). The following variables were considered:

- Number of peptides per protein (*n*), obtained from the *peptide-to-protein* integration;
- Protein log2 ratios (*Xinf*), calculated from the *protein-to-proteinall* integration;
- Protein grand mean (*Xsup*), calculated from the *protein-to-proteinall* integration;
- *FDR* values indicating which *Z* values stand out as outliers in the *protein-to-proteinall* integration;
- Species information (*tags*) showing in the *protein-to-proteinall* integration output file.



Sample folder(s)	Lower level	Higher level	Reported vars	Output report	Headers of columns to eliminate	Merge with report
*	peptide	protein	n	Npep2prot		
*	protein	proteinall	Xinf, Xsup, Z , FDR , tags	Npep2prot_Quanprot	peptide	Npep2prot

Figure 65. The REPORT module task table for workflow 4.

The following advanced options of the REPORT module have been used in this workflow (*Figure 65*):

- *Merge with report* designates the report file whose variables will be incorporated into the current report on the basis of their element identifiers. Thus, the *Npep2prot_Quanprot* report is first populated with the *Xinf, Xsup, FDR* and *tags* values of the protein-to-proteinall integration, and then added the number of peptides per protein coming from the *Npep2prot report* (*Figure 65*). This is possible because the *protein* level is present in both reports;
- *Column headers to eliminate* restricts the elements to be written to the corresponding *Output report* to those from the *protein* level (lower level; both lower and higher levels are used unless otherwise stated). This prevents data duplication when *Npep2prot_Quanprot* and *Npep2prot* reports are merged, as the latter contains also the peptide level.

In agreement with the results reported [6], the combined statistics *B_vs_A* analysis shows human, yeast and bacterial proteins normally distributed around the expected 0, 1, and -2 mean-corrected log2-ratio (X_{qa} – *grand mean*, indeed, they are the X_{inf} - X_{sup}) values (corresponding to 1, 2, and 0.25 fold change values) (*Figure 66*).

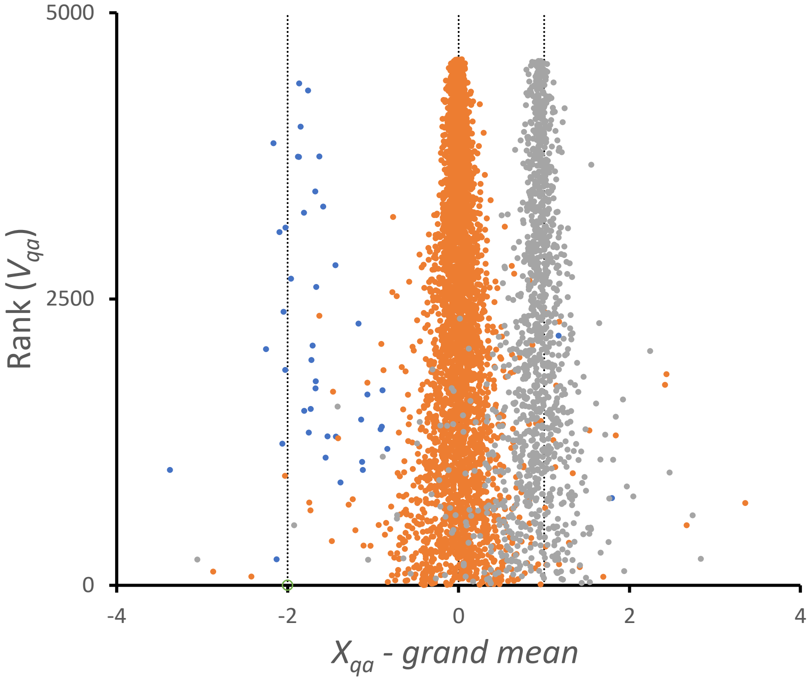


Figure 66. Quantification of human (orange), yeast (grey) and bacterial (blue) proteins according to the combined statistics B_vs_A.

Workflow execution

The workflow template and input files that are needed to execute this workflow can be downloaded from https://github.com/CNIC-Proteomics/iSanXoT/wiki/studies/cases/templates/WPP_LabelFree.zip

See the *Executing a workflow template* Section below for detailed instructions.

Importing a workflow template

To illustrate how to execute the workflows created with iSanXoT, we shall make use of the sample workflows described in detail in the previous Section. Start by downloading, for instance, workflow 1 template and input files from the iSanXoT wiki (https://github.com/CNIC-Proteomics/iSanXoT/wiki/studies/workflows/WSPP-SBT.zip). Extract the files included in the compressed archive Then proceed as follows:

- Open the iSanXoT application by double-clicking the application icon (*Figure 67*).

Figure 67. The iSanXoT startup message.

- Choose *New Project* from the *Project* menu (*Figure 68*).

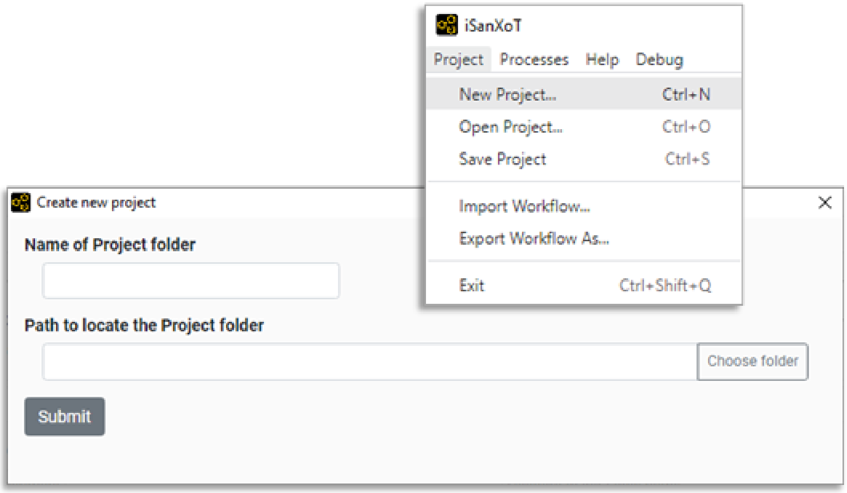


Figure 68. Create New Project.

- Provide a project name and a project folder, then click the *Submit* button (*Figure 68*).
- To import a workflow previously generated with iSanXoT, choose *Import Workflow* from the *Project* menu (*Figure 69*). Choose, for instance, workflow 1 by selecting the root folder where you stored the workflow files.

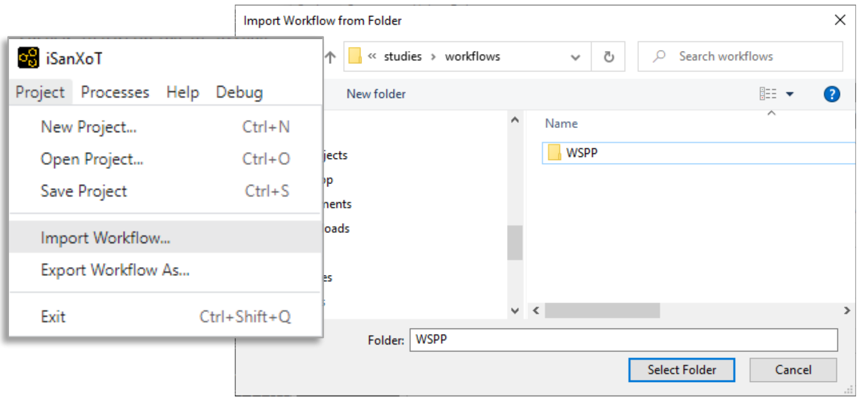
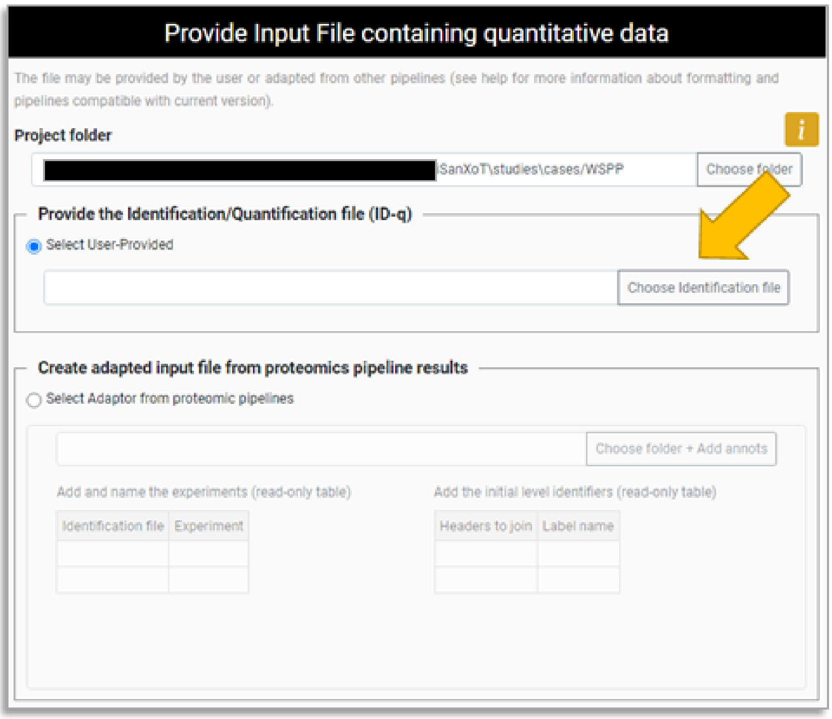


Figure 69. Importing a preexisting iSanXoT workflow to the newly-created project.

- Now click on *Choose identification file* to provide an identification/quantification (*ID-q*) file (*Figure 70*). The below section shows how to prepare this file based on the output from a variety of proteomics pipelines; but you can try the *ID-q.tsv* file included in the root folder where you stored workflow 1 files.



Provide Input File containing quantitative data

The file may be provided by the user or adapted from other pipelines (see help for more information about formatting and pipelines compatible with current version).

Project folder

[redacted]SanXoT\studies\cases/WSPP

Choose folder

Provide the Identification/Quantification file (ID-q)

● Select User-Provided

Choose Identification file

Create adapted input file from proteomics pipeline results

○ Select Adaptor from proteomic pipelines

Choose folder + Add annots

Add and name the experiments (read-only table)

Identification file	Experiment

Add the initial level identifiers (read-only table)

Headers to join	Label name

Figure 70. Choosing the identification/quantification (ID-q) file for the newly-created project.

- Bear in mind that any path included in the RELS CREATOR module task table of the workflow imported will need be updated.

Creating the identification/quantification file from proteomics pipelines

iSanXoT requires an identification/quantification file (*ID-q.tsv*) containing identified features together with their quantitative values and the experiment they originate from whether there is more than one experiment. This tab-separated values *ID-q* file must fulfill:

- Features (scan, peptide, protein, etc.) are unequivocally referenced using a unique identifier;
- Quantitative values for the features are displayed across columns;
- Experiments are specified in a column field headed by *Experiment* whether there is more than one experiment.

In this Section we shall describe how to prepare the *ID-q.tsv* file based on data obtained with the four most popular proteomics pipelines.

Table 1 . Column headers from the tested proteomic pipelines that are needed for the input idnetification/quantification file (ID-q.tsv). (*) Redundant proteins. You have to select your main protein. (**) Provide the experiment names in the case of multiple experiments.

			Id_q.txt column headers				
			Features			**Quantitation**	**Experiment**
Proteomic pipeline	**Experiment type**	**File type**	**Scan**	**Peptide**	**Protein**		
Proteome Discoverer (version 2.5)	label free	*_PSMs.txt*	Spectrum file + First Scan + Charge	Sequence + Modifications	Protein Accessions (*)	Intensity	Experiment names (**)
	isotopically labelled	*_PSMs.txt*	Spectrum file + First Scan + Charge	Sequence + Modifications	Protein Accessions (*)	Abundance: Channel identifier	Experiment names (**)
MaxQuant (version 1.6.5.0)	label free	*peptides.txt*	Id	Sequence	Leading razor protein	Intensity sample	Experiment names (**)
	isotopically labelled	*msmsScans.txt*	Raw file + Scan number + Charge	Sequence + Modifications	Proteins (*)	Reporter intensity sample	Experiment names (**)
Trans-Proteomics Pipeline (version 6.0.0)	label free	*pepXML* file	spectrum	peptide	proteins (*)	precursor_intensity	Experiment names (**)
	isotopically labelled	*pepXML* file	spectrum	peptide	proteins (*)	Channel identifier	Experiment names (**)
Fragpipe (version 1.8.1)	label free	*psm.tsv*	Spectrum	Modified Peptide	Protein	Intensity	Experiment names (**)
	isotopically labelled	*psm.tsv*	Spectrum	Modified Peptide	Protein	Channel identifier	Experiment names (**)

Preparing the *ID-q* file from Proteome Discoverer output

In the case of Proteome Discoverer version 2.5 [9], the quantitative data required to prepare the *ID-q* file are extracted by the *Processing* workflow nodes *Reporter Ions Quantifier* (for labelled experiments) and *Minora Feature Detector* (for label-free experiments). These data are written by the *Consensus* workflow to a text file suffixed with *_PSMs.txt* displaying:

Label-free experiments

- *PSMs Peptide ID*: Unique identifier for each *Peptide* sequence identified;
- *Sequence*: Amino acid sequence of the identified peptide;
- *Modifications*: Post-translational modifications contained within the sequence;
- *Protein Accessions*: Accession code(s) for the protein(s) to which the peptide sequence is ascribed;
- *Intensity:* Precursor ion intensity (to be used in label-free experiments).

The unequivocally identifier for a peptide has to be composed by the union of *Sequence* and *Modifications* fields.

Labelled experiments

- *Spectrum File*: Name of the RAW file the spectral MS/MS data was extracted from;
- *First Scan*: RAW file derived scan number for the MS/MS spectrum;
- *Charge*: Charge state of the precursor ion;
- *PSMs Peptide ID*: Unique identifier for each *Peptide* sequence identified;
- *Sequence*: Amino acid sequence of the identified peptide;
- *Modifications*: Post-translational modifications contained within the sequence;
- *Protein Accessions*: Accession code(s) for the protein(s) to which the peptide sequence is ascribed;
- *Abundance: Channel identifier*: Reporter ion intensity (to be used in labelled experiments).

In the case of multiple experiments, we recommend the creation of a unique identifier for a Scan is composed by the union of the fields: *Spectrum File*, *First* Scan and *Charge*; otherwise, the *PSMs Peptide ID* field is enough for the Scan identification. The unequivocally identifier for a peptide has to be composed by the union of *Sequence* and *Modifications* fields.

Preparing the *ID-q* file from MaxQuant output

The way MaxQuant version 1.6.5.0 [8] data are adapted for use with iSanXoT depends on whether they originate from label-free or labelled proteomics experiments:

Label-free experiments

In this case, the necessary data can be found in the *peptides.txt* file, which is stored in the *"…combined/txt*" folder:

- *Id*: Unique, consecutive identifier for each peptide *Sequence* identified;

- *Sequence*: Amino acid sequence of the identified peptide;
- *Leading razor protein*: Identifier of the top scoring protein to which the peptide *Sequence* is ascribed;
- *Intensity + Experiment*: Summed up extracted ion current of all isotopic clusters associated with the peptide *Sequence* identified in the raw files making up *Experiment*.

Labelled experiments

When dealing with labelled experiments (e.g. iTRAQ- or TMT-based), the necessary data can be found in the *msmsScans.txt* file, which is stored in the *"…combined/txt*" folder:

- *Raw file*: Name of the RAW file from which the MS/MS data were extracted;
- *Scan number*: MS/MS spectrum number within a given RAW file;
- *Charge*: Charge state of the precursor ion;
- *Sequence*: Amino acid sequence of the identified peptide. This parameter is nonblank only when identification was successful.
- *Proteins*: Identifier(s) of the protein(s) to which the peptide *Sequence* is ascribed;
- *Reporter intensity + n*: Intensity of the reporter ion corresponding to sample *n*.

The *msmsScans.txt* file lacks the unique identifier that is required in the *ID-q* file. The user is advised to create this unique identifier by concatenating *RAW file, Scan number,* and *Charge*.

Preparing the *ID-q* file from Trans-Proteomic Pipeline output

Trans-Proteomics Pipeline version 6.0.0 [10] comprises a number of tools (*e.g. ASAPRatio*, *XPRESS* and *Libra*) that allow both label-free and isotopically labelled quantitative data to be saved to a *pepXML* file. Using the *PepXML Viewer* application, this file can be exported as a spreadsheet displaying:

Label-free experiments

- *Spectrum*: Unique identifier for each *Peptide* sequence identified;
- *Peptide:* Amino acid sequence of the identified peptide;
- *Protein:* Identifier of the protein to which the *Peptide* sequence is ascribed;
- *Precursor_intensity:* Precursor ion intensity (to be used for label-free quantification).

Labelled experiments

- *Spectrum*: Unique identifier for each *Peptide* sequence identified;
- *Peptide:* Amino acid sequence of the identified peptide;
- *Protein:* Identifier of the protein to which the *Peptide* sequence is ascribed;
- *Channel identifier*: Reporter ion intensity (to be used for isotopic labelling quantification).

Preparing the *ID-q* file from FragPipe output

Fragpipe version 1.8.1 [11] relies on the *Quant (MS1)* and *Quant (Isobaric)* modules to generate a *psm.tsv* output file that contains the quantitative data obtained from label-free and labelled experiments, respectively:

Label-free experiments

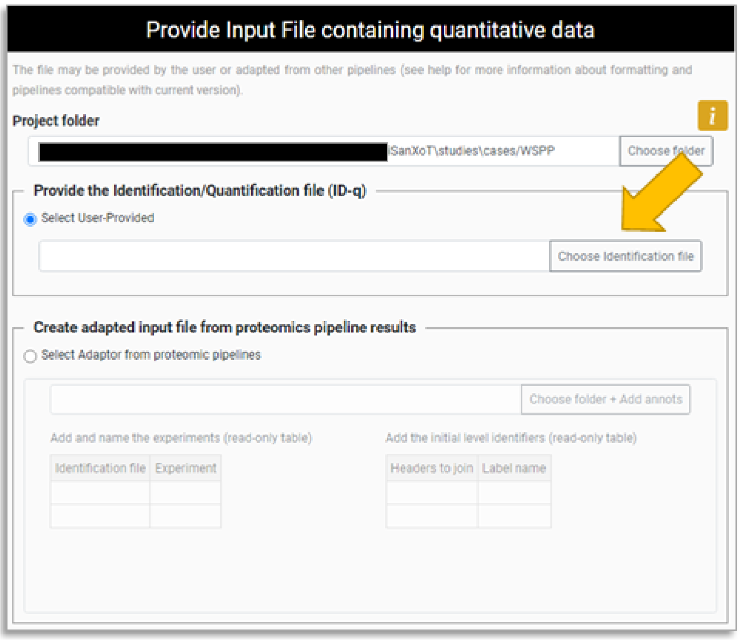
- *Spectrum*: Unique identifier for each *Peptide* sequence identified;
- Modified Peptide: Peptide sequence including modifications, modified residues;
- *Protein:* Identifier of the protein to which the *Peptide* sequence is ascribed;
- *Intensity:* Precursor ion intensity (to be used in label-free experiments);
- *Channel identifier*: Reporter ion intensity (to be used in labelled experiments).

Labelled experiments

- *Spectrum*: Unique identifier for each *Peptide* sequence identified;
- Modified Peptide: Peptide sequence including modifications, modified residues;
- *Protein:* Identifier of the protein to which the *Peptide* sequence is ascribed;
- *Intensity:* Precursor ion intensity (to be used in label-free experiments);
- *Channel identifier*: Reporter ion intensity (to be used in labelled experiments).

Adapting the results from proteomics pipelines for iSanXoT

iSanXoT requires an identification/quantification tab-separated values file (*ID-q.tsv*) containing identified features together with their quantitative values and the experiment they originate from. Users can either compose this *ID-q* file manually (see Section above to learn how to do that with data from the four most popular proteomics pipelines) or have it prepared by the iSanXoT Input Adaptor. The latter option is described in this Section.



Provide Input File containing quantitative data

The file may be provided by the user or adapted from other pipelines (see help for more information about formatting and pipelines compatible with current version).

Project folder

[redacted]iSanXoT\studies\cases/WSPP

Provide the Identification/Quantification file (ID-q)

● Select User-Provided

Create adapted input file from proteomics pipeline results

○ Select Adaptor from proteomic pipelines

Add and name the experiments (read-only table)

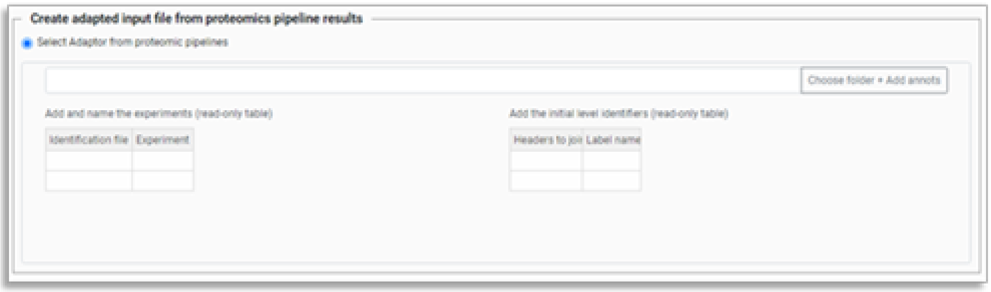
Identification file	Experiment

Add the initial level identifiers (read-only table)

Headers to join	Label name

Figure 71. Selecting an ID-q file in the Input Adaptor main window.

- Open the iSanXoT application, then see Section above to learn how to get to the *Input Adaptor* main window (*Figure 71*).
- If you already have an appropriate *ID-q* file, select the *Provide the Identification/Quantification file (ID-q)* option, then click on *Choose identification file* to select the file.



Create adapted input file from proteomics pipeline results

● Select Adaptor from proteomic pipelines

Add and name the experiments (read-only table)

Identification file	Experiment

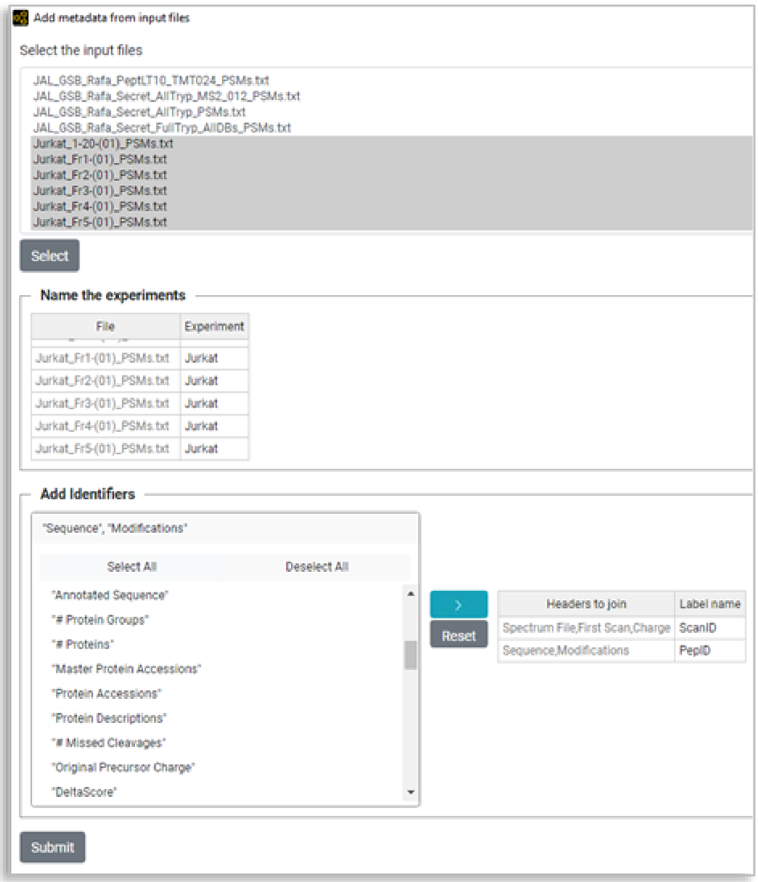
Add the initial level identifiers (read-only table)

Headers to join	Label name

Figure 72. Having the iSanXoT Input Adaptor prepare the ID-q file.

- To have the iSanXoT Input Adaptor prepare the *ID-q* file from proteomics pipelines results (this works at least with MaxQuant, Trans-Proteomic Pipeline, FragPipe, and

Proteome Discoverer), select the *Create adapted input file from proteomics pipelines results* option, then click on *Choose folder + Add anots* to select the folder containing the output files (*Figure 72*). A three-panel window will pop-up (*Figure 73*).



Add metadata from input files

Select the input files

JAL_GSB_Rafa_PeptLT10_TMT024_PSMs.txt
JAL_GSB_Rafa_Secret_AllTryp_MS2_012_PSMs.txt
JAL_GSB_Rafa_Secret_AllTryp_PSMs.txt
JAL_GSB_Rafa_Secret_FullTryp_AllDBs_PSMs.txt
Jurkat_1-20-(01)_PSMs.txt
Jurkat_Fr1-(01)_PSMs.txt
Jurkat_Fr2-(01)_PSMs.txt
Jurkat_Fr3-(01)_PSMs.txt
Jurkat_Fr4-(01)_PSMs.txt
Jurkat_Fr5-(01)_PSMs.txt

Name the experiments

File	Experiment
Jurkat_Fr1-(01)_PSMs.txt	Jurkat
Jurkat_Fr2-(01)_PSMs.txt	Jurkat
Jurkat_Fr3-(01)_PSMs.txt	Jurkat
Jurkat_Fr4-(01)_PSMs.txt	Jurkat
Jurkat_Fr5-(01)_PSMs.txt	Jurkat

Add Identifiers

"Sequence", "Modifications"

"Annotated Sequence"
"# Protein Groups"
"# Proteins"
"Master Protein Accessions"
"Protein Accessions"
"Protein Descriptions"
"# Missed Cleavages"
"Original Precursor Charge"
"DeltaScore"

Headers to join	Label name
Spectrum File,First Scan,Charge	ScanID
Sequence,Modifications	PepID

Figure 73. Adapting results from a proteomics pipeline. In the top panel several output files from Proteome Discoverer have been selected. These PSMs.txt *files, which contain identification/quantification data, have been assigned an experiment name* (Jurkat) *in the middle panel. The bottom panel has been used to create identifiers by concatenating result file headers:* ScanID *(by concatenating* Spectrum File, First Scan *and* Charge*) and* pepID *(by concatenating* Sequence *and* Modifications*).*

- The top panel displays the files included in the folder so that you can select one or more result files to be considered by iSanXoT Input Adaptor. Please bear in mind that if several result files are selected, these must necessarily have the same column headers.
- The middle panel is used to set the distribution of data items across experiments according to result filenames.
- The bottom panel allows to create identifiers by concatenating result file headers. It is composed of two interfaces:
    - The headers found on the result files are listed on the left. Header names will be added to the interface on the right as they are selected;
    - The interface on the right displays the header names selected to generate the identifier as well as the identifier name provided by the user.

- Please note that the alphanumeric text that unambiguously identifies the items to be integrated is the only identifier that must be necessarily included in the *ID-q* file.

- Click the *Submit* button and the Input Adaptor will start generating the *ID-q.tsv* file.

References

[1] Trevisan-Herraz M, Bagwan N, Garcia-Marques F, Rodriguez JM, Jorge I, Ezkurdia I, et al. SanXoT: a modular and versatile package for the quantitative analysis of high-throughput proteomics experiments. Bioinformatics. 2019;35(9):1594-6.

[2] Navarro P, Trevisan-Herraz M, Bonzon-Kulichenko E, Nunez E, Martinez-Acedo P, Perez-Hernandez D, et al. General statistical framework for quantitative proteomics by stable isotope labeling. J Proteome Res. 2014;13(3):1234-47.

[3] Garcia-Marques F, Trevisan-Herraz M, Martinez-Martinez S, Camafeita E, Jorge I, Lopez JA, et al. A Novel Systems-Biology Algorithm for the Analysis of Coordinated Protein Responses Using Quantitative Proteomics. Mol Cell Proteomics. 2016;15(5):1740-60.

[4] Bonzon-Kulichenko E, Camafeita E, Lopez JA, Gomez-Serrano M, Jorge I, Calvo E, et al. Improved integrative analysis of the thiol redox proteome using filter-aided sample preparation. J Proteomics. 2020;214:103624.

[5] Gonzalez-Amor M, Garcia-Redondo AB, Jorge I, Zalba G, Becares M, Ruiz-Rodriguez MJ, et al. Interferon stimulated gene 15 pathway is a novel mediator of endothelial dysfunction and aneurysms development in angiotensin II infused mice through increased oxidative stress. Cardiovasc Res. 2021.

[6] Navarro P, Kuharev J, Gillet LC, Bernhardt OM, MacLean B, Rost HL, et al. A multicenter study benchmarks software tools for label-free proteome quantification. Nat Biotechnol. 2016;34(11):1130-6.

[7] Navarro P, Vazquez J. A refined method to calculate false discovery rates for peptide identification using decoy databases. J Proteome Res. 2009;8(4):1792-6.

[8] Tyanova S, Temu T, Cox J. The MaxQuant computational platform for mass spectrometry-based shotgun proteomics. Nat Protoc. 2016;11(12):2301-19.

[9] Orsburn BC. Proteome Discoverer-A Community Enhanced Data Processing Suite for Protein Informatics. Proteomes. 2021;9(1).

[10] Deutsch EW, Mendoza L, Shteynberg D, Slagel J, Sun Z, Moritz RL. Trans-Proteomic Pipeline, a standardized data processing pipeline for large-scale reproducible proteomics informatics. Proteomics Clin Appl. 2015;9(7-8):745-54.

[11] Kong AT, Leprevost FV, Avtonomov DM, Mellacheruvu D, Nesvizhskii AI. MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. Nat Methods. 2017;14(5):513-20.