User Guide:
Help and documentation
(1.2.8)

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 (v1.2.8):

https://github.com/CNIC-Proteomics/iSanXoT/wiki/docs/user_guides/User_Guide_iSanXoT-1.2.8.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:

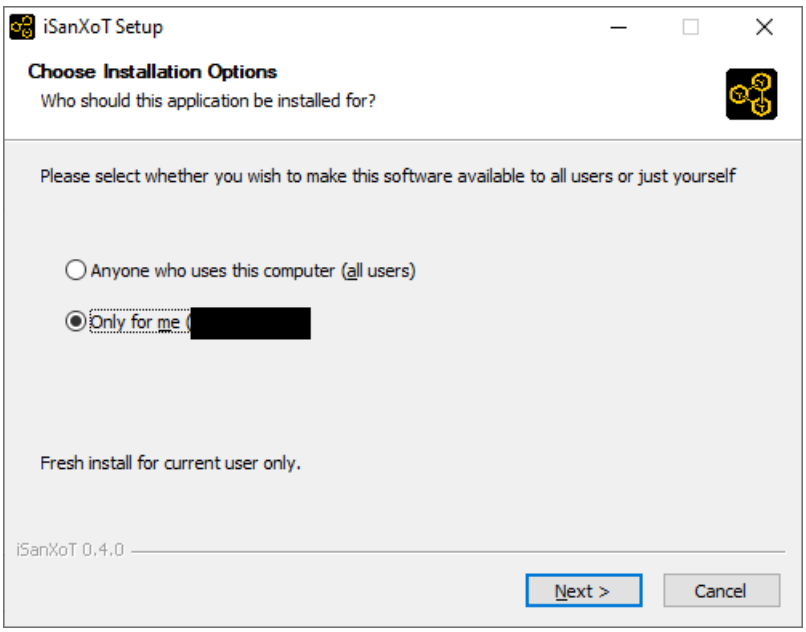
- Windows 10 Pro (x64)
- MacOs High Sierra (10.13.6)
- Ubuntu 20.04 (x64)

Windows distribution

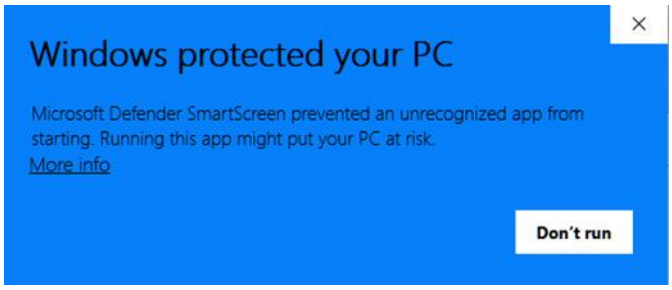
The iSanXoT Windows distribution is packaged in a NSIS Launcher (exe file).

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

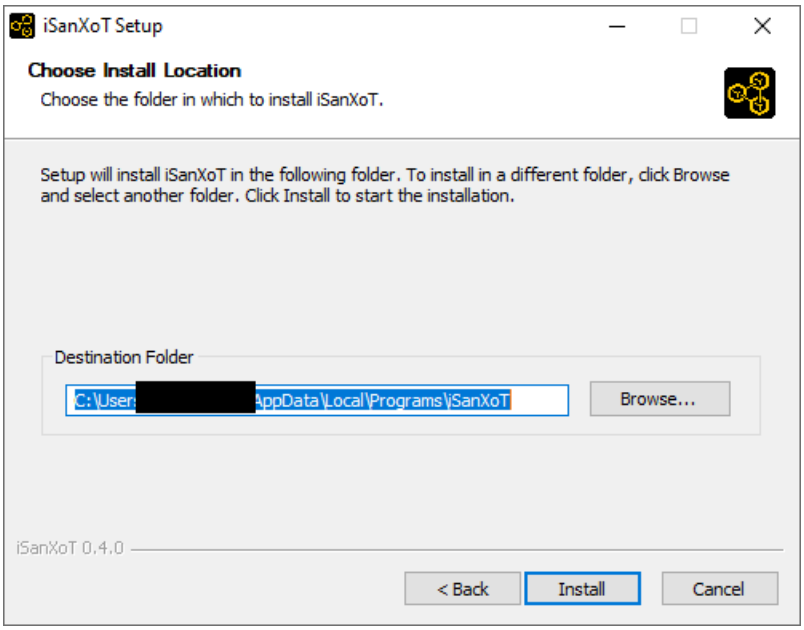
Double-click the Launcher file; the Installer window will show up:



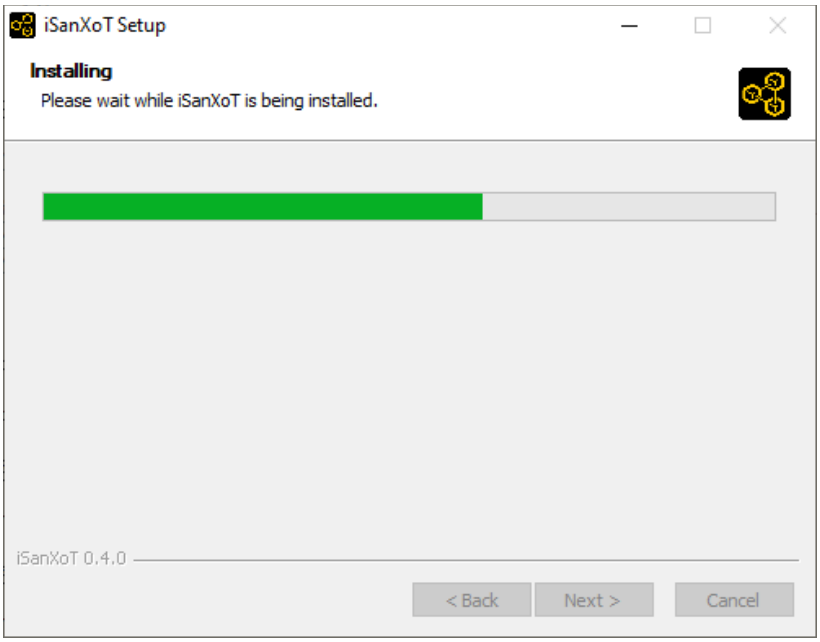
WARNING: Windows Defender SmartScreen might show a prompt suggesting that you cancel the installation; in such case click “More info” and then select the “Run anyway” option.



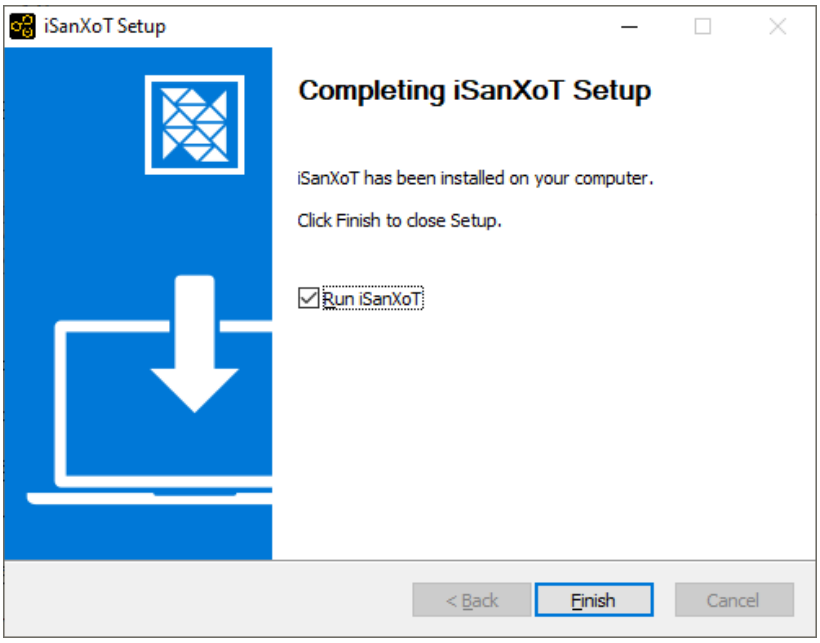
Then, you can choose the iSanXoT installation folder:



Wait while iSanXoT is being installed.



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

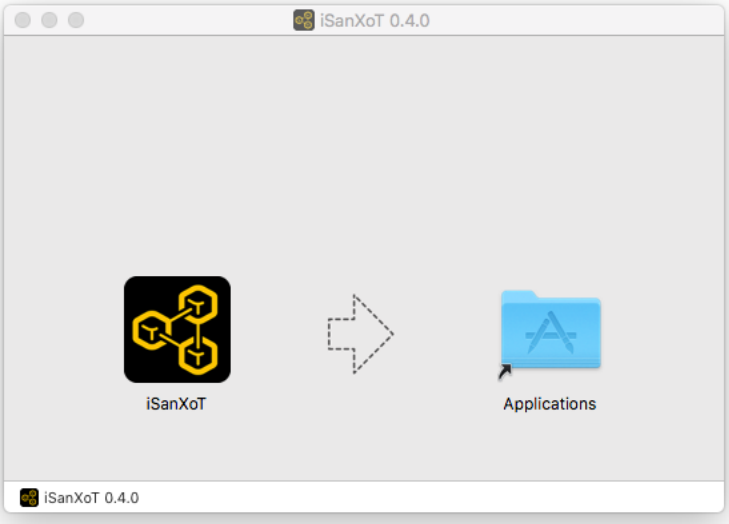


MacOS distribution

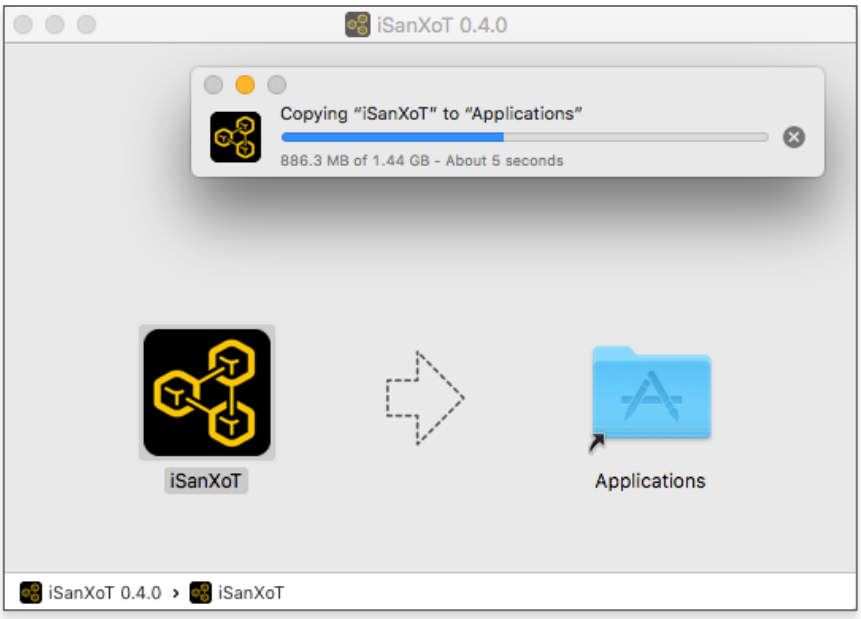
The iSanXoT MacOS distribution is packaged in a DMG container.

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...

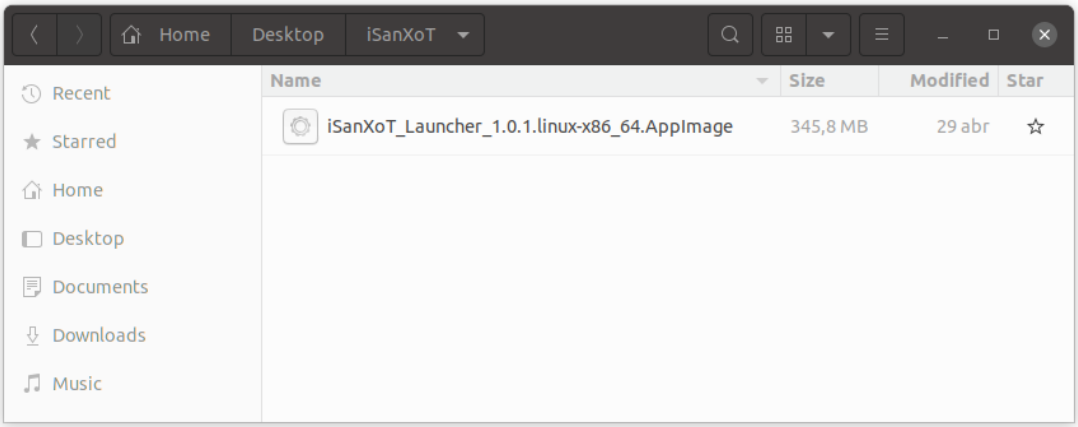


and you're done: the iSanXoT application is now installed.

Linux distribution

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

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



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.

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.

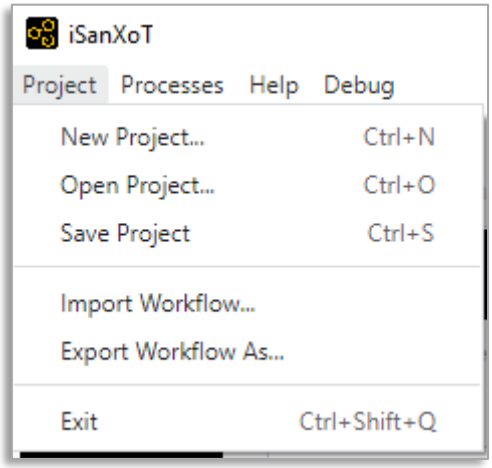


Figure 2. Project menu.

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.

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.

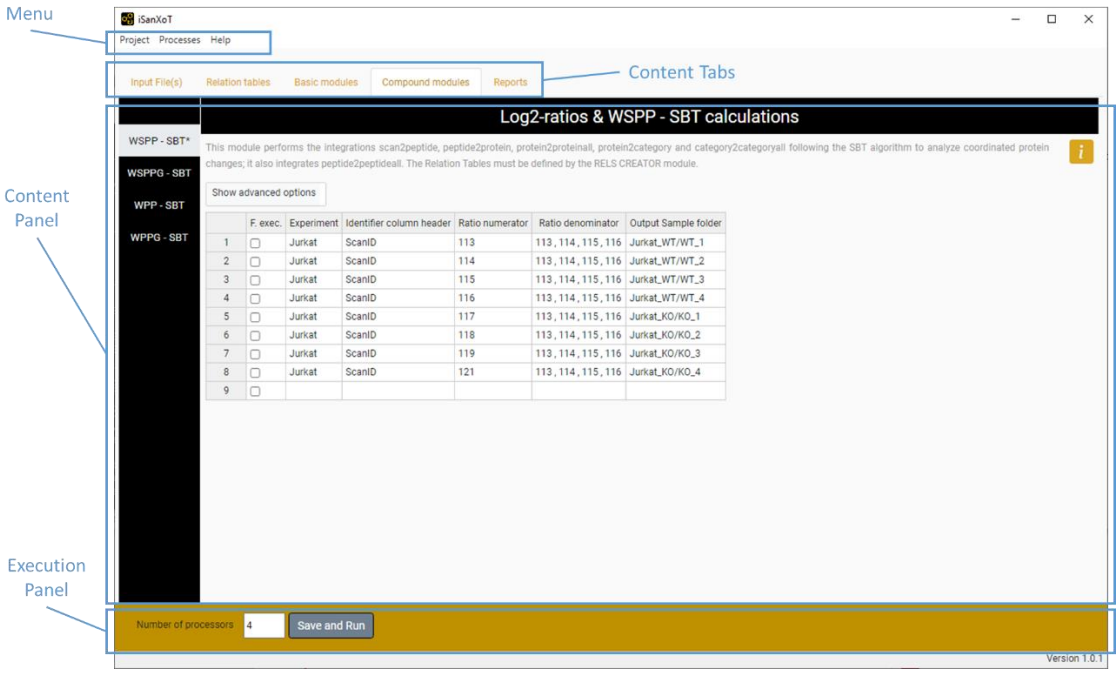


Figure 4. Main View of iSanXoT.

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.

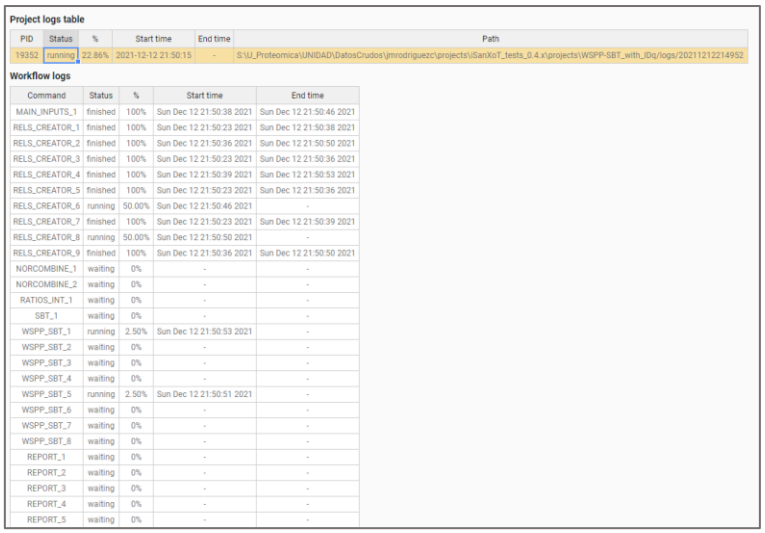
As well as the “.isanxot” folder, the following folders are necessary to shape your project:

- 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).

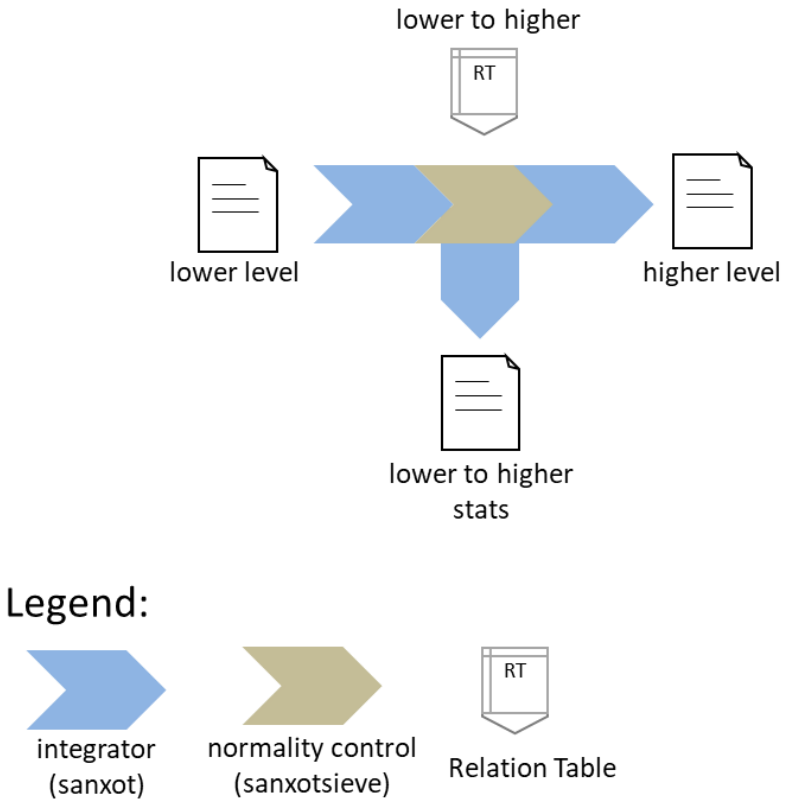


Figure 10. Schematic representation of the INTEGRATE module. *The integration is carried out from any lower level to any higher level using the programs "SanXoT" and "SanXoTSieve" and the generic integration algorithm (GIA).*

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.

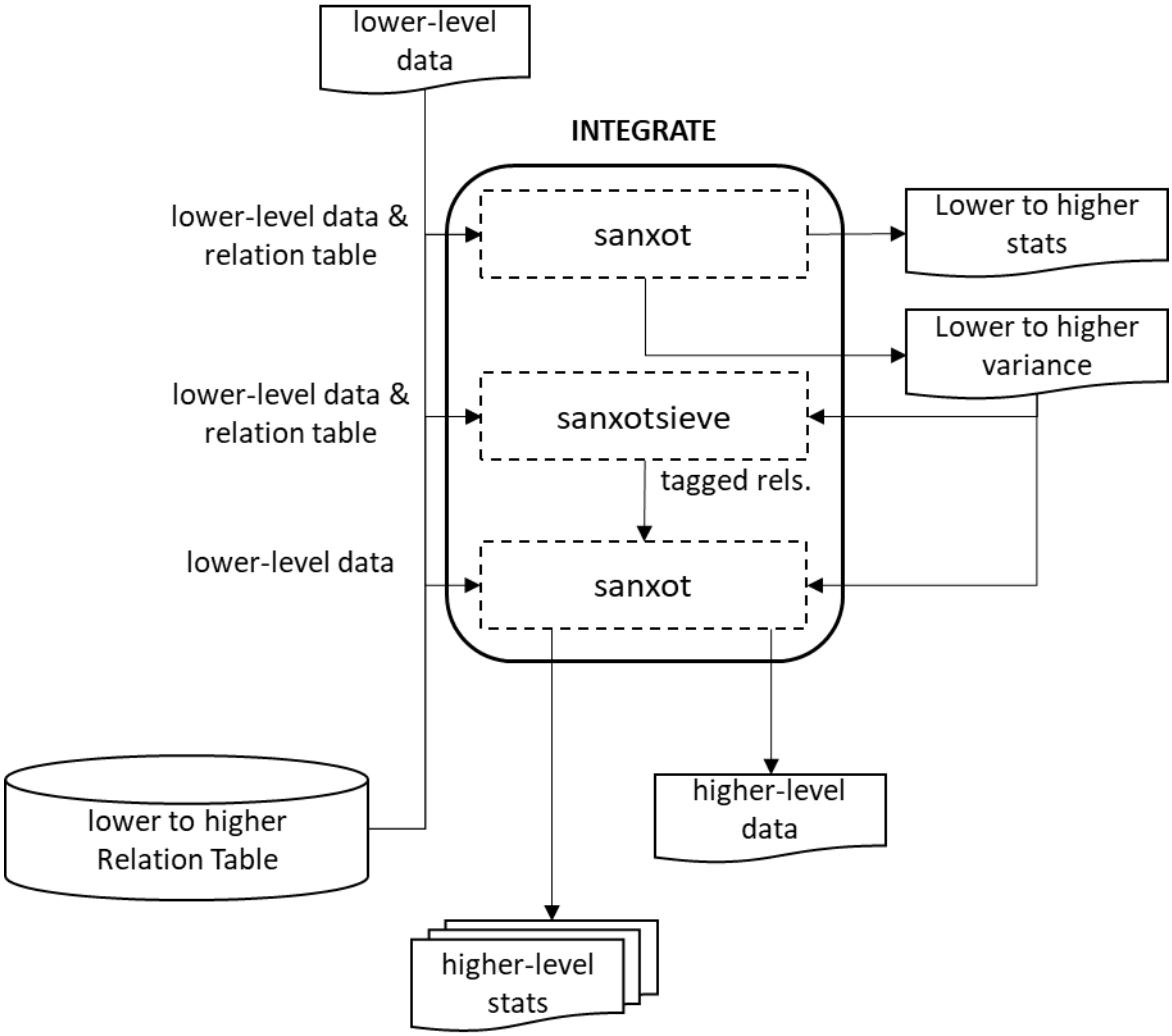


Figure 11. The INTEGRATE module flowchart. *A first integration is done with "SanXoT" that calculates the variance; then "SanXoTSieve" removes outliers tagging them in a new relation table; finally, a second integration is done with "SanXoT" using the variance calculated.*

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.

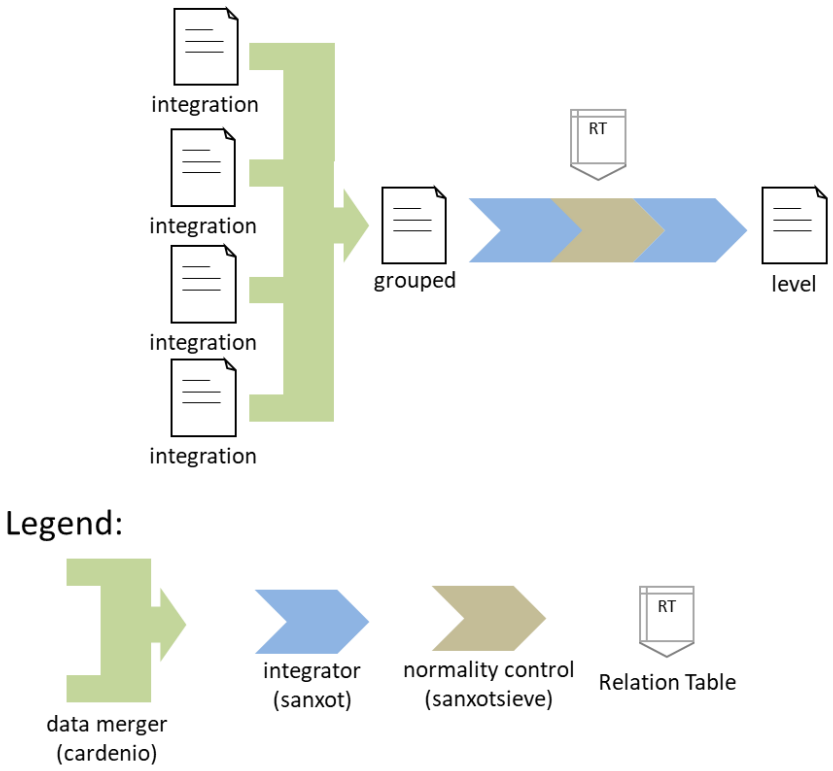


Figure 14. Schematic representation of the NORCOMBINE module used to combine technical or biological replicates.

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**).

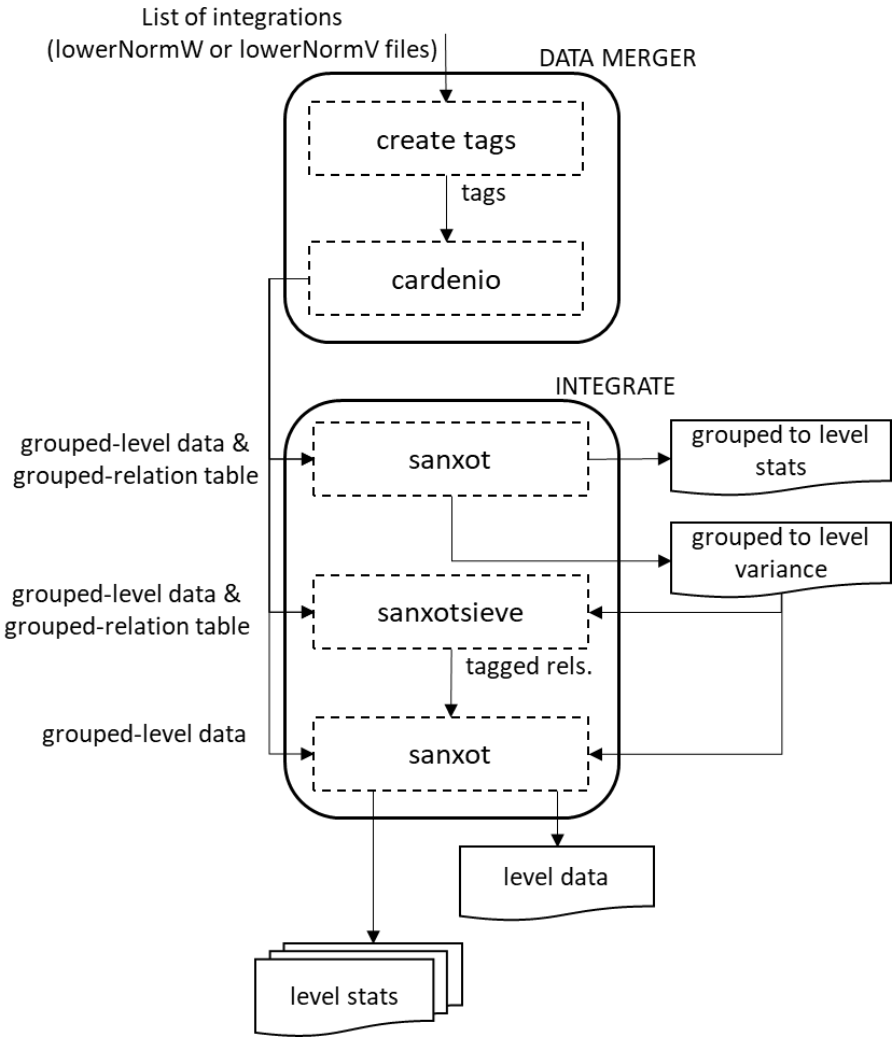


Figure 15. The NORCOMBINE module flowchart.

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.

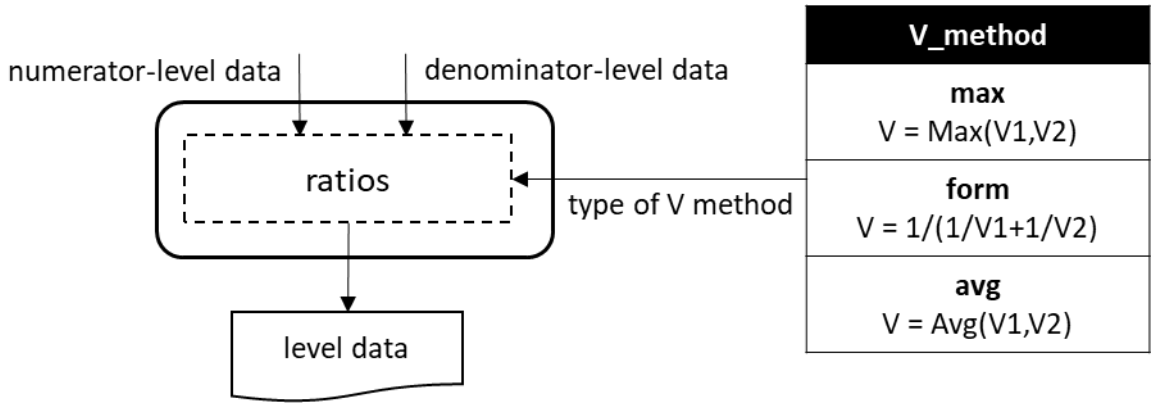


Figure 18. The RATIOS module flowchart.

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.

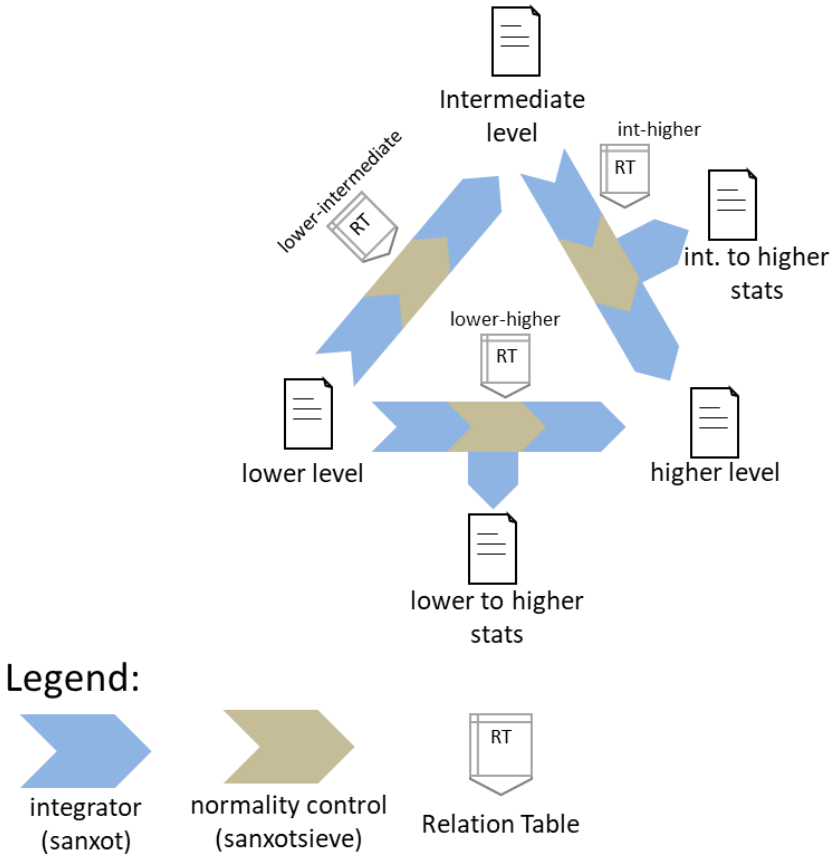


Figure 20. Schematic representation of the SBT module.

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.

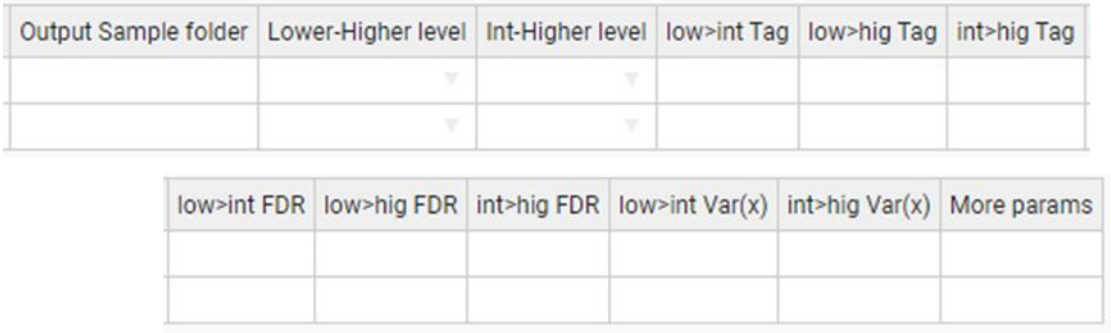


Figure 22. A sample task table with advanced parameters in the SBT module.

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	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	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

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 in REPORT module

The REPORT module module accepts 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)

(FDR_category2category < 0.05) & (Z_protein2proteinall >= 2 | Z_protein2proteinall <= -2)

Filter in SANSON module

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 protein or peptide abundance changes in a variety of biological contexts. Note that these workflows may be easily reused to process new data (see next section).

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 this workflow. This study characterizes the molecular alterations that take place along time when vascular smooth muscle cells (VSMCs) are treated with angiotensin-II (AngII) for 0, 2, 4, 6, 8, and 10 h. Quantitative proteomics was performed using isobaric iTRAQ 8-plex labeling. Workflow 1 analyzes a) protein abundance changes and b) functional category alterations produced by the coordinated behaviour of proteins at each one of the times, in relation to time 0. This is done using, in only one step, the compound module WSPP-SBT, which performs automatically all the required tasks.

Workflow operation

Workflow 1 requires the RELS CREATOR module, the WSPP-SBT compound module and the REPORT basic module (*Figure 35*). The relation tables required to perform the integrations are created by the RELS CREATOR module (*Figure 35*A) from a table provided by the user. The WSPP-SBT module performs a sequence of consecutive integrations based on the WSPP statistical model [2] and the SBT algorithm [3] (*Figure 35*B). Finally, the REPORT module organizes the data in tables containing the information needed.

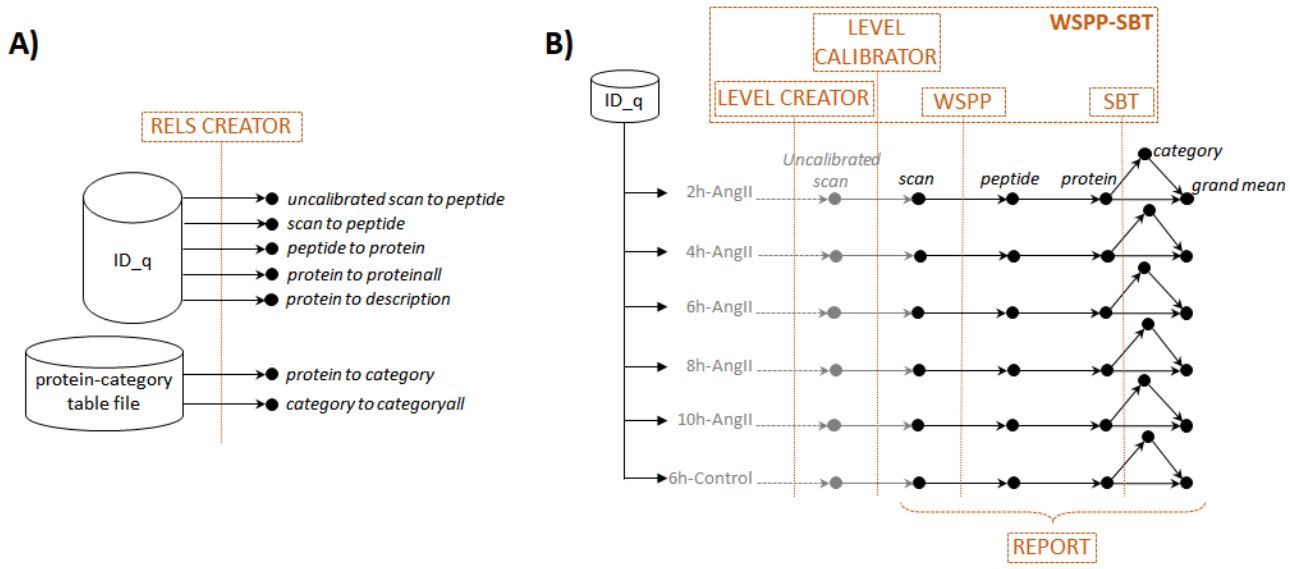


Figure 35. Scheme of workflow 1 (one-step quantification in a labeled experiment) showing module components: RELS CREATOR (A) and WSPP-SBT and REPORT (B)

The WSPP-SBT module needs the user to define the meaning of relative abundances, which iSanXoT always expresses as log2ratios. In this case the abundance data corresponds to the intensities of iTRAQ reporters at the scan level, which are tabulated in the "ID-q" file with the name of each reporter as column header (see below how these tables are generated). The intensities of each scan at 0 h are in column "Abundance: 113" and are used as a common reference to express abundance ratios and are therefore used as denominator. The reporter intensities corresponding to the different time points are used as numerators of the ratios. The task table also allows the user to put an easily identifiable name in the folders where the quantitative values of each sample are stored (*Figure 36*).

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

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

The WSPP-SBT module first performs a calibration to assign a statistical weight to each one of the log2ratio values at the scan level (*Figure 35*B), as described [2]. The statistical weight of each scan is the inverse of the estimated variance associated with the log2 of intensity ratios [2]. Once the data is calibrated at the scan level, the workflow performs the integrations *scan-to-peptide* and *peptide-to-protein*.

At the protein level the SBT algorithm is then applied for the detection of functional category changes originated by the coordinated behaviour of proteins (*Figure 35*B). The algorithm first calculates the variance of the *protein-to-category* integration, which is an improved estimate of the technical protein variance, since it is less influenced by biological changes [3]. This protein variance is used to perform the *protein-to-grand mean* integration (hereinafter referred to as *protein-to-proteinall*) integration, from which statistically significant abundance changes are detected. The algorithm finally performs the *category-to-grand mean* integration (hereinafter referred to as *category-to-categoryall*), from which statistically significant category changes are detected. All the results from the integrations executed by the WSPP-SBT module are saved, per each sample, to the *Output Sample folder* indicated in the module task table (*Figure 36*).

Every integration step needs a relation table (a text file) that links lower- to higher-level elements. Relation tables can be automatically created by the RELS CREATOR module (*Figure 35*A, upper) and can be provided by the user (*Figure 35*A, lower). In this example (*Figure 37*) the relation tables linking scan to peptides and peptides to proteins are obtained from the "ID-q" file, just by indicating the name of the columns where they are located (in this case "Scan_Id", "Pep_Id" and "Master Protein Accessions"). In this case the columns *Master Protein Accessions* and *Master Protein Descriptions* in the "ID-q" file contain the accession numbers and the complete name of the proteins, respectively, so that a relation table *protein2description* is also created; this relation table may be later used to append the full name of the protein to any of the created reports (see below). An example of the *peptide2protein* relation table, linking the peptides identified to the proteins they come from is showed in *Figure 38*A. The elements of the relation table protein2category were retrieved from a text file containing functional annotations for mouse proteins gathered from several protein function databases (*Figure 38*B), as described by the authors [3]. Note that relation tables are by default extracted from the ID-q file; to use other text files the absolute path with the location of the text file has to be indicated. The relation tables *protein2proteinall* and *category2categoryall* guide the integration to a grand mean (a common element called "[1]"). The integration *peptide2peptideall* is not necessary in this workflow but is included in this example since it may be useful to inspect quantifications at peptide level.

Relation Table to be created	Column name of Lower level	Column name of Higher level	Column name of 3rd c	Table from which RT is extracted
uscan2peptide	Scan_Id	Pep_Id		
scan2peptide	Scan_Id	Pep_Id		
peptide2protein	Pep_Id	Master Protein Accessions		
peptide2peptideall	Pep_Id	[1]		
protein2proteinall	Master Protein Accessions	[1]		
protein2description	Master Protein Accessions	Master Protein Descriptions		
protein2category	Protein	Category		{PATH}/DAVID_IPA_merged_noDups_feb14_IJ.txt
category2categoryall	Category	[1]		{PATH}/DAVID_IPA_merged_noDups_feb14_IJ.txt

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

A)

protein	peptide
P23242	KVAAGHELQPLALVDQRPSSR__N-Term(iTRAQ8plex);K1(iTRAQ8plex)
P17182	AAVPSGASTGLYEALELRDNDKTR__N-Term(iTRAQ8plex);K22(iTRAQ8plex)
P23780	AGATLDLLVENMGR__N-Term(iTRAQ8plex)
A2AIM4; P58771-2; P58774-2; E9Q45(	RLQLVEEELDRAQER__N-Term(iTRAQ8plex)
P20152	EKLQEEMLQREEAESTLQSFR__N-Term(iTRAQ8plex);K2(iTRAQ8plex)
P99024	MAVTFLGNSTALQELFKR__N-Term(iTRAQ8plex);M1(Oxidation);K17(iTRAQ8plex)
A1BN54; Q9JI91; P57780; Q7TPR4	KHEAFESDLAAHQDR__N-Term(iTRAQ8plex);K1(iTRAQ8plex)
Q91YQ5	ASSFVLALEPELESR__N-Term(iTRAQ8plex)
O08547	NLGSLNTELQDVQR__N-Term(iTRAQ8plex)
Q9EQ06	SVAGELVLLTGAGHGLGR__N-Term(iTRAQ8plex)
P05064; Q9CPQ9	FSNEELAMATVTALR__N-Term(iTRAQ8plex)
P10126	KDGSASGTTLLEALDCLLPPTRPTDKPLR__N-Term(iTRAQ8plex);C16(Carbamidomethyl);K26(iTRAQ8plex)

B)

category	protein
DAVID_PANTHER_BP_ALL_BP00141:Transport	Q9Z351
DAVID_PANTHER_BP_ALL_BP00142:Ion transport	Q9Z351
DAVID_PANTHER_BP_ALL_BP00143:Cation transport	Q9Z351
DAVID_PANTHER_PATHWAY_P00042:Muscarinic acetylcholine receptor 1 and 3 s	Q9Z351
DAVID_SP_PIR_KEYWORDS_alternative splicing	Q9Z351
DAVID_SP_PIR_KEYWORDS_ion transport	Q9Z351
DAVID_SP_PIR_KEYWORDS_ionic channel	Q9Z351
DAVID_KEGG_PATHWAY_mmu04020:Calcium signaling pathway	Q9Z329
DAVID_KEGG_PATHWAY_mmu04070:Phosphatidylinositol signaling system	Q9Z329
DAVID_KEGG_PATHWAY_mmu04114:Oocyte meiosis	Q9Z329
DAVID_KEGG_PATHWAY_mmu04270:Vascular smooth muscle contraction	Q9Z329

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

Once the integrations are performed, the REPORT module is used to collect from the *Output sample folders* stated by the user the statistical variables desired and to organize them in tables (*Figure 39*). In this case, the results from the samples (2h-AngII, 4h-AngII, 6h-AngII, 8h-AngII, and 10h-AngII) are to be tabulated.

In this example the REPORT module creates a protein table and a category table by performing the following steps:

- Create a table called “Npep2prot" containing the number of peptides with which each protein is quantified.
    - This is done by extracting from the *peptide-to-protein* integrations in the indicated folders the number of elements (n) of the lower level (peptide) used to quantitate the higher level (protein).

- Create a table called “Npep2prot_Quantprot_filtered” containing the protein changes Zqa and the statistical significance FDRqa of these changes.
    - This is done by extracting from the *protein-to-proteinall* integration in the indicated folders the standardized log2 ratios (Z) and False Discovery Rates (FDR) of the lower level (protein).
- Add to this table the number of peptides with which each protein is quantified.
    - This is done by merging the previous table with the existing table “Npep2prot” according to the level common to the two tables (protein), without including a specific column (peptide) and eliminating replicate entries.
- Add to this table an additional column with the complete description of the proteins.
    - This is done by merging the previous table with the relation table protein2description according to the level common to the two tables (protein).
- Filter the table so that only the proteins having a statistically significant abundance change (FDR < 0.01) are tabulated.

    - This is done by applying in the Filter column a condition based on the FDR to the results from the protein2proteinall integration. For more detailed information, see the "Filter for report" in the iSanXoT wiki: https://github.com/CNIC-Proteomics/iSanXoT/wiki.

- Create a table called "Nprot2cat" containing the number of proteins with which each category is quantified.
    - This is done by extracting from the *protein-to-category* integrations in the indicated folders the number of elements (n) of the lower level (protein) used to quantitate the higher level (category).

- Create a table called "Nprot2cat_Quantcat_filtered" containing the category changes Zca and the statistical significance FDRca of these changes.
    - This is done by extracting from the *category-to-categoryall* integration in the indicated folders the standardized log2 ratios (Z) and False Discovery Rates (FDR) of the lower level (category).
- Add to this table the number of proteins with which each category is quantified.
    - This is done by merging the previous table with the existing table "Nprot2cat" according to the level common to the two tables (category), without including a specific column (protein) and eliminating replicate entries.
- Filter the table so that only the categories having a statistically significant change (FDR < 0.01) are tabulated.
    - This is done by applying in the Filter column a condition based on the FDR to the results from the category2categoryall integration.

- Create a table called "Npep2prot_Quanprot" containing the number of peptides per protein, the protein changes Zqa and the statistical significance FDRqa of these changes.
    - This is done as explained above, omitting the protein descriptions and the filters.

- Create a table called "Nprot2cat_Quancat_Quanprot_filtered" containing the category changes Zca and the statistical significance FDRca of these changes.
    - This is done by extracting from the *category-to-categoryall* integration in the indicated folders the standardized log2 ratios (Z) and False Discovery Rates (FDR) of the lower level (category).
- Add to this table the number of proteins per category, the protein changes Zqa and the statistical significance FDRqa of these changes.
    - This is done by merging the previous table with the existing tables "Nprot2cat" and "Npep2prot_Quantprot".
- Filter the table so that only the categories containing 5 or more proteins or 100 or less proteins are tabulated.
    - This is done by applying in the Filter column a set of conditions joined with the "&" operator.

Note that these commands in the REPORT module, which allow to construct tables required in typical quantitative proteomics projects, are easily reusable for other projects.

Sample folder(s)	Lower level	Higher level	Reported vars	Output report	Column headers to eliminate	Merge with report	Add columns from relation tal	Filter
2h-AngII , 4h-AngII , 6h-AngII , 8h-AngII , 10h-AngII	peptide	protein	n	Npep2prot				
2h-AngII , 4h-AngII , 6h-AngII , 8h-AngII , 10h-AngII	protein	proteinall	Z , FDR	Npep2prot_Quanprot_filtered	peptide	Npep2prot	protein2description	FDR_protein2proteinall < 0.01
2h-AngII , 4h-AngII , 6h-AngII , 8h-AngII , 10h-AngII	protein	category	n	Nprot2cat				
2h-AngII , 4h-AngII , 6h-AngII , 8h-AngII , 10h-AngII	category	categoryall	Z , FDR	Nprot2cat_Quancat_filtered	protein	Nprot2cat		FDR_category2categoryall < 0.01
2h-AngII , 4h-AngII , 6h-AngII , 8h-AngII , 10h-AngII	protein	proteinall	Z , FDR	Npep2prot_Quanprot	peptide	Npep2prot		
2h-AngII , 4h-AngII , 6h-AngII , 8h-AngII , 10h-AngII	category	categoryall	Z , FDR	Nprot2cat_Quancat_Quanprot_filtered		Nprot2cat , Npep2prot_Quanprot		(n_protein2category >=5) & (n_protein2category <= 100)

Figure 39. The REPORT task table for workflow 1.

In *Figure 41* we show two heat maps constructed from the protein and category tables obtained with the REPORT module.

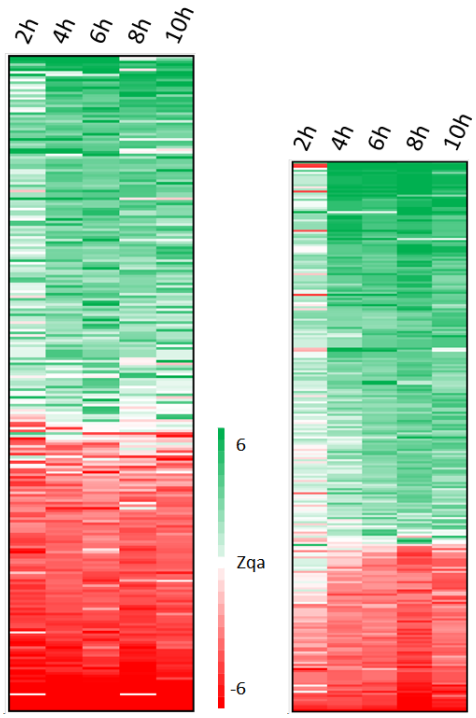


Figure 40. Relative abundance changes of proteins (Zqa, Left) and functional categories (Zca, Right) generated from the "Npep2prot_Quanprot_filtered" and "Nprot2cat_Quancat_filtered" reports, respectively, obtained by the REPORT module of workflow 1 (Figure 39). Both report tables were sorted by the average of Zqa and Zca, respectively.

In Figure 41 we show some examples of functional categories showing statistically significant changes produced by coordinated protein behaviour, plotted using the data in the "Nprot2cat_Quancat_Quanprot_filtered" table generated by the REPORT module.

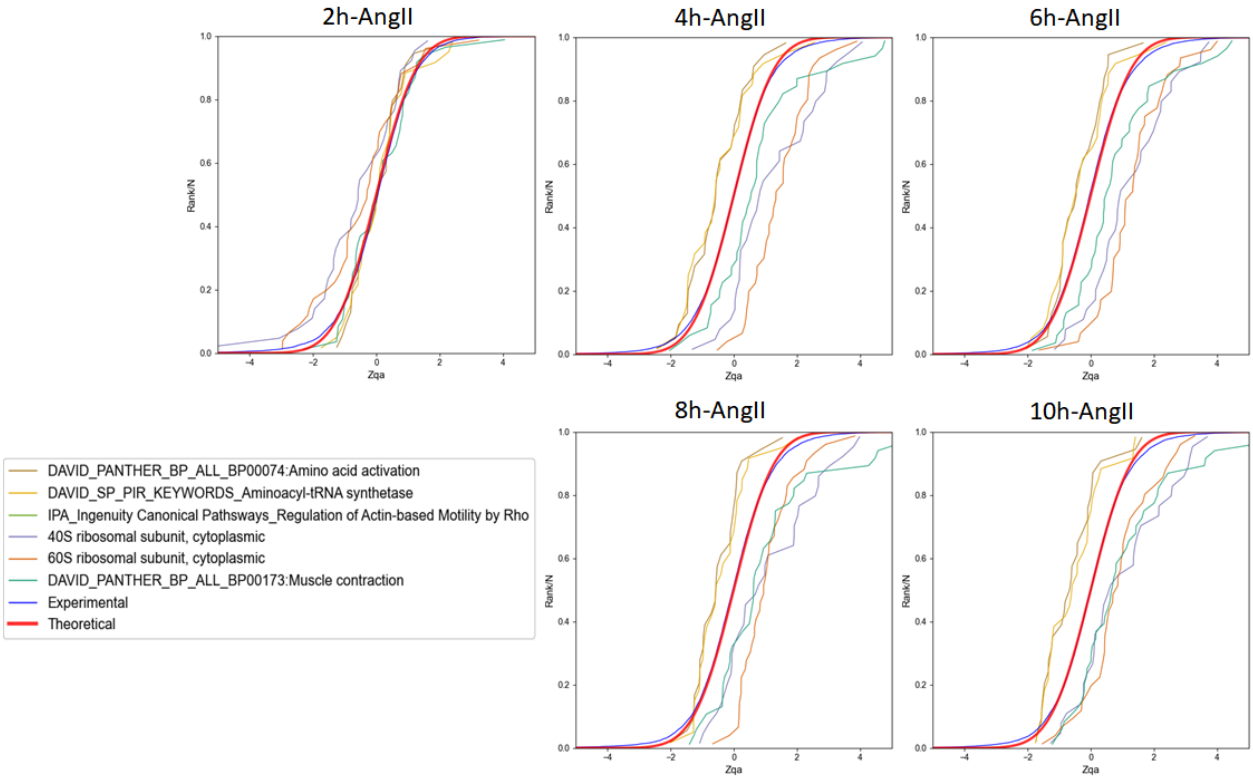


Figure 41. Examples of time-dependent coordinated protein behavior of VSMCs treated with angiotensin-II as revealed by the distribution of the standardized log2 ratio (Zqa) of protein components in each category.

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 *Importing a workflow template* Section below for detailed instructions.

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

Experimental

The data from González-Amor *et al*. [5] was used to illustrate this workflow. This study analyzes the contribution of interferon-stimulated gene 15 (ISG15) to the vascular damage associated with hypertension, using KO mutants for this gene and subjecting or not the animals to AngII treatment. This experiment contained 16 samples from mouse aortic tissue corresponding to four groups: four WT-Control mice, four ISG15-KO mice, four WT+AngII mice, and four ISG15-KO+AngII mice. The experiment was performed using two isobaric iTRAQ 8-plex batches. This module illustrates how to create, step by step, a workflow to integrate the quantitative results from each sample to the protein level, to integrate protein data from the four biological replicates in each group, to construct ratios between two conditions, and to analyze functional category changes due to coordinated protein behavior using the SBT model.

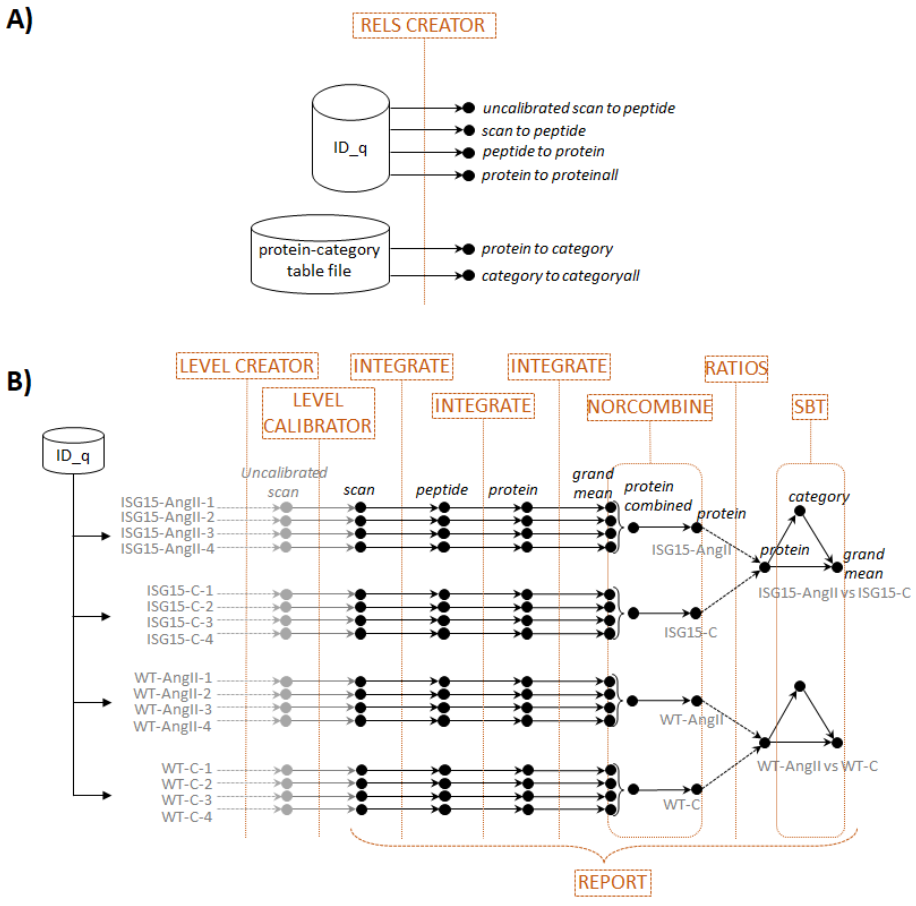


Figure 42. Scheme of workflow 2 (step-by-step quantification and sample combination in a labeled experiment) showing module components: RELS CREATOR (A) and LEVEL CREATOR, LEVEL CALIBRATOR, INTEGRATE, NORCOMBINE, RATIOS, SBT, and REPORT (B)

Workflow operation

Workflow 2 comprises all six basic modules: LEVEL CREATOR, LEVEL CALIBRATOR, INTEGRATE, NORCOMBINE, RATIOS, and SBT as well as the REPORT module (*Figure 42*). The task table in the starting module, LEVEL CREATOR, generates the files at the scan level containing the log2 ratios and the corresponding sample folders (*Figure 43*). In this example, as in workflow 1, the name of each iTRAQ reporter was used as column header in the "ID-q" file containing the intensities. In addition, the column Experiment indicates whether the intensities come from the first or second iTRAQ 8-plex batch. Each iTRAQ batch contains two biological replicates from each of the four groups and the average of reporter intensities from the two untreated WT mice (reporters in the columns "113" and "117") were used as internal control within each batch, and therefore were used as denominator for the log2ratios.

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 43. The LEVEL CREATOR task table for workflow 2.

LEVEL CREATOR generates the *u_scan* (*uncalibrated scan*) files, which contain the scan identifiers (taken from the column "Scan_Id" in the "ID-q" table), the log2-ratios at the scan level Xs (as defined in the task table) and the *uncalibrated* weights Vs (which in the WSPP model are the intensities of the reporters in the Ratio numerator column) (*Figure 44*). The uncalibrated weights Vs are related to the quality of quantification (a higher weight implicates a more accurate quantification), but they are not associated to a statistical variance yet.

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 44. 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).*

The LEVEL CALIBRATOR module calibrates the Vs weights by performing a *u_scan-to-peptide integration* and generates the *scan* (*calibrated scan*) files, which contain true, calibrated statistical weights (defined in the WSPP model as the inverse of the estimated individual scan variances) (*Figure 45*, Top).

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

Note that LEVEL CALIBRATOR automatically generates a plot to supervise the accuracy of calibrations ("*_outGraph_VRank" png file) in each sample folder, which show whether the model is able to predict experimental scan variances as a function of the calibrated statistical weights (*Figure 46*).

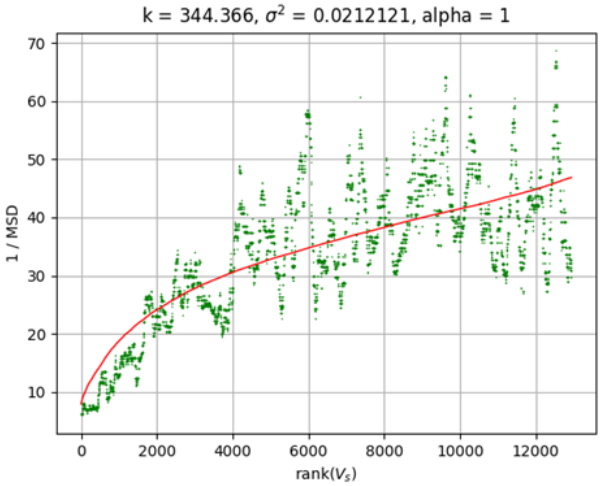


Figure 46. Automatically generated graphs to supervise the accuracy of calibrations. These graphs represent 1/MSD versus the rank of Vs (scan weight, which in this level corresponds to reporter intensity). MSD is the Mean Squared Deviation of the scans vs the respective mean of the peptide they belong to. The scans are ordered by Vs and the MSD is calculated in a sliding window of 200 scans [2].

As in the case of workflow 1, before performing the integrations, the relation tables have to be created with the RELS CREATOR module, which have a similar structure (*Figure 47*).

Relation Table to be created	Column name of Lower level	Column name of Higher level	Column na	Table from which RT is extracted
u_scan2peptide	Scan_Id	Pep_Id		
scan2peptide	Scan_Id	Pep_Id		
peptide2protein	Pep_Id	Master Protein Accessions		
protein2proteinall	Master Protein Accessions	[1]		
protein2category	Protein	Category		{PATH}/uniprot-MusMusculus_GOanotationDavid_enero2018_rels_v8_Protein2Category.txt
category2categoryall	Category	[1]		{PATH}/uniprot-MusMusculus_GOanotationDavid_enero2018_rels_v8_Protein2Category.txt

Figure 47. The RELS CREATOR task table for workflow 2.

The INTEGRATE module performs the *scan-to-peptide*, *peptide-to-protein* and *protein-to-proteinall* integrations according to the module task table (*Figure 45*, Bottom). Note that for consistency all the files created for each sample are stored by default in the folder indicated in the task table of LEVEL CREATOR module, unless otherwise indicated in the Output Sample folder column.

Note that the INTEGRATE module automatically generates a plot to check the accuracy of the GIA integration model in each one of the integration steps. This is done by comparing the distribution of Z values with that of the null hypothesis (standard normal distribution) (see *Figure 49* below, left panels). These graphs are stored in the corresponding sample folders (in "*_outGraph.png" files).

By default, iSanXoT removes integration outliers. To prevent the removal of outlier elements in the *protein-to-proteinall* integration, as these are just the proteins which are significantly altered, a *0 FDR* value was indicated in the INTEGRATE task table for this integration (Figure 45, Bottom).

Once protein levels are created, workflow 2 uses the NORCOMBINE basic module (*Figure 42*B) to integrate protein values from the four biological replicates in each group to produce integrated protein values per group that are stored in the folders WT-C, WT-AngII, ISG15-C and ISG15-AngII (*Figure 48*).

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 48. The NORCOMBINE task table for workflow 2.

The NORCOMBINE module integrates biological replicates within sample groups applying the GIA algorithm [3], which models the distribution of protein values around the average taking into account error propagation theory and estimates a global variance for the integration. The GIA algorithm assumes that the individual variances of all the lower elements (proteins) are affected by a global variance (which in this case arises from biological variability within the same group). While this assumption may not hold in all the cases, it can be easily checked by inspecting the test distributions. The NORCOMBINE module (like the INTEGRATE module) automatically generates graphs comparing the distribution of the integrated Z variables with those of the standard normal distribution. As shown in *Figure 49* right, the distribution of protein Z values estimated by the model in the case of the ISG15-AngII group agree very well with the null hypothesis, demonstrating that the assumption of the model is a good approach to treat the biological variance of the samples within this group. Similar results were obtained in the other three groups (not shown).

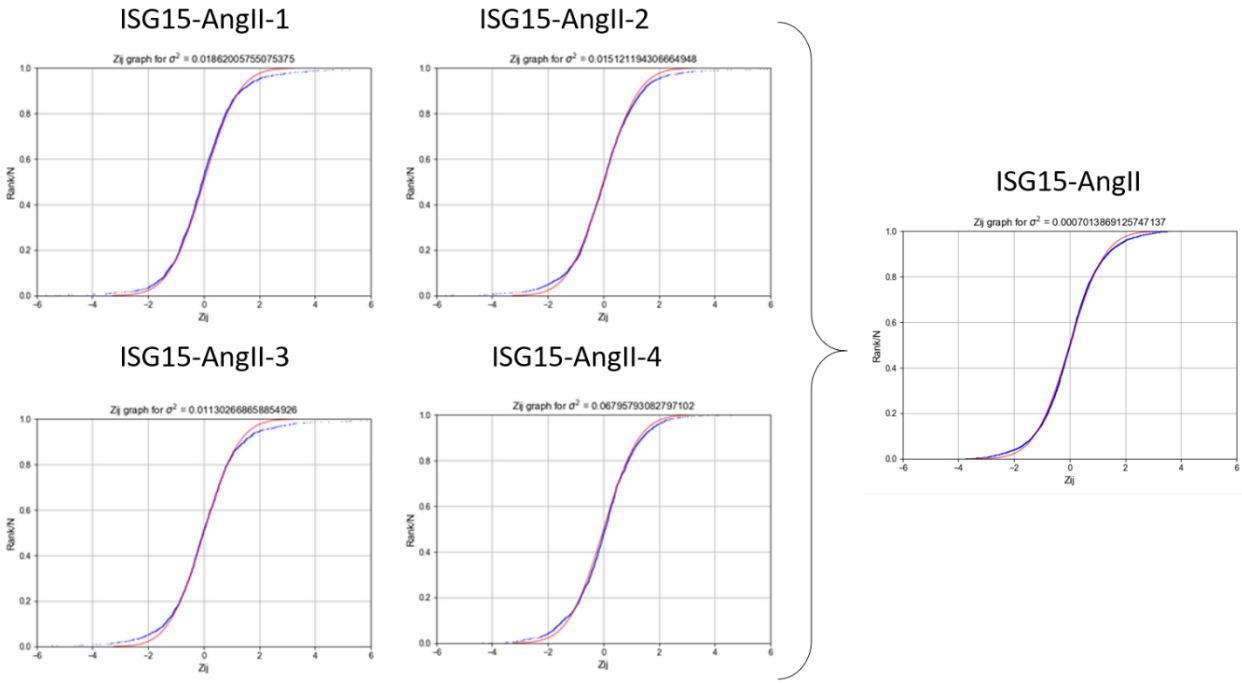


Figure 49. Distribution of the standardized log2 protein ratios (Zqa) from the four individual ISG15-AngII VSMC samples (Left panel) showing how the WSPP model agrees well with the

expected null distributions in the four cases, and from the integrated ISG15-AngII sample group obtained with the NORCOMBINE module (Right panel), showing how the GIA assumption of a global biological variance is a good approach to treat the biological variability of samples within this group. Red: null hypothesis (standard distribution); blue: experimental data.

Note also that the NORCOMBINE module performs a weighted averaging from several samples and that from the good fitting to the null hypothesis it allows an accurate control over outliers (*Figure 49*). This unique approach allows the integration of protein values originating from unbalanced sample groups, distinct experiments or mass spectrometers, and even different labelling techniques (see for instance [2]).

The module task table indicates that samples were combined at the *protein* level using the *proteinall* level for normalization. This means that log2 protein ratios are firstly normalized by the grand mean before being integrated into an averaged protein value; this compensates for differences in protein load into each iTRAQ channel. Note that proteins could also be integrated to other levels (for instance organules, subcellular compartments, complexes, …), before being integrated by NORCOMBINE, allowing different kinds of normalizations. Finally, the column lowerNorm indicates which is the file that contains the normalized data, which usually are the *lowerNormV* files, previously generated by the INTEGRATE module. Further details can be found in the iSanXoT documentation.

The protein averages from the four biological 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 ISG15 controls (*Figure 50*). The V method column allows the user to indicate the method used to assign a statistical weight to the log2ratios, which by default is the method called "max" (for further details see "RATIOS" module in the iSanXoT wiki, https://github.com/CNIC-Proteomics/iSanXoT/wiki).

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 50. The RATIOS task table for workflow 2.

The last basic module executed in workflow 2, SBT (*Figure 51*), applies the SBT algorithm to the above defined comparisons for the detection of functional category changes originated by the coordinated behaviour of proteins. This is made as explained in workflow 1. The SBT module is more flexible since it allows to perform the triangle operations to any kind of level, not only proteins. In this case the triangle is made by the levels protein and category (*Figure 50*) and the corresponding grand mean.

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

Figure 51. The SBT task table for workflow 2.

Finally, the REPORT module (*Figure 52*) is used as in workflow 1 to generate tables with the protein and category data. In this case additional features of the REPORT module are used. The table “Npep2prot" is generated using an asterisk. This symbol is used by iSanXoT as a wildcard character to indicate that the results from all the samples containing the *peptide-to-protein* integration (i.e. ISG15-AngII-1, ISG15-AngII-2, ISG15-AngII-3, ISG15-AngII-4, ISG15-AngII, ISG15-C-1, ISG15-C-2, ISG15-C-3, ISG15-C-4, ISG15-C, and ISG15-AngIIvsISG15-C) are to be included in the table. However, the “Npep2prot_Quanprot_ISG15_filtered” and “Npep2prot_Quanprot_WT_filtered” tables include the protein changes (Zqa), the statistical significance (FDRqa) of these changes, and the number of peptides per protein only from the samples indicated in the *Sample folder(s)* column. The report for the ISG15 samples is filtered by Zqa to show the most extreme values (greater than 1 or less than -1) but only for the “ISG15-AngIIvsISG15-C” sample. Additional filters for the minimum number of peptides per protein are also used in these tables. The tables containing category values are filtered by Zca (greater than or equal to 2 or less than or equal to -2) and/or by the number of proteins per category (between 5 and 100).

Sample folder(s)	Lower level	Higher level	Reported vars	Output report	Column headers to elimi	Merge with report	Ad	Filter
*	peptide	protein	n	Npep2prot				
ISG15-AngII-1 , ISG15-AngII-2 , ISG15-AngII-3 , ISG15-AngII-4 , ISG15-AngII , ISG15-C-1 , ISG15-C-2 , ISG15-C-3 , ISG15-C-4 , ISG15-C , ISG15-AngIIvsISG15-C	protein	proteinall	Z , FDR	Npep2prot_Quanprot_ISG15_filtered	peptide	Npep2prot		(Z_protein2proteinall@ISG15-AngIIvsISG15-C <= -1 | Z_protein2proteinall@ISG15-AngIIvsISG15-C >= 1) & (n_peptide2protein >= 2)
WT-AngII-1 , WT-AngII-2 , WT-AngII-3 , WT-AngII-4 , WT-AngII , WT-C-1 , WT-C-2 , WT-C-3 , WT-C-4 , WT-C , WT-AngIIvsWT-C	protein	proteinall	Z , FDR	Npep2prot_Quanprot_WT_filtered	peptide	Npep2prot		(n_peptide2protein > 4)
ISG15-AngIIvsISG15-C , WT-AngIIvsWT-C	protein	category	n	Nprot2cat				
ISG15-AngIIvsISG15-C , WT-AngIIvsWT-C	category	categoryall	Z , FDR	Nprot2cat_Quancat_filtered	protein	Nprot2cat		(Z_category2categoryall <= -2 | Z_category2categoryall >= 2) & (n_protein2category >= 5) & (n_protein2category <= 100)
ISG15-AngIIvsISG15-C , WT-AngIIvsWT-C	protein	proteinall	Z , FDR	Nprot2cat_Quanprot		Nprot2cat		
ISG15-AngIIvsISG15-C , WT-AngIIvsWT-C	category	categoryall	Z , FDR	Nprot2cat_Quancat_Quanprot_filtered		Nprot2cat_Quanprot		(n_protein2category >= 5) & (n_protein2category <= 100)

Figure 52. The REPORT task table for workflow 2.

The tables generated by REPORT can be used to generate heatmaps showing the most relevant protein abundance changes (*Figure 53*). As previously shown [5], iSanXoT analysis revealed a coordinated alteration of proteins implicated in cardiovascular function, extracellular matrix and remodeling, and vascular redox state in aortic tissue from AngII-infused ISG15-KO mice (*Figure 54A*). The coordinated protein behavior from some of the altered categories can be analyzed in the sigmoid plots (*Figure 54B*).

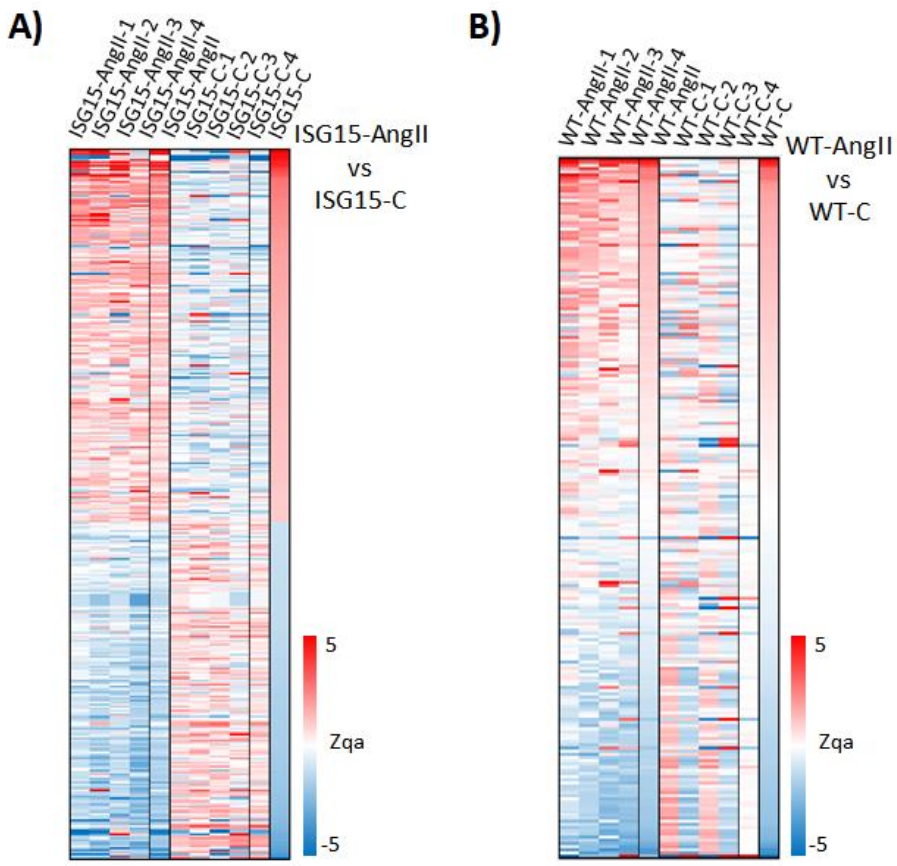


Figure 53. Differential abundance of functional proteins revealed by workflow 2. The heatmap (A) for proteins (Zqa) is based on the "Npep2prot_Quanprot_ISG15_filtered" REPORT table. The heatmap (B) displays the proteins (Zqa) for the WT samples using the "Npep2prot_Quanprot_WT_filtered" REPORT table.

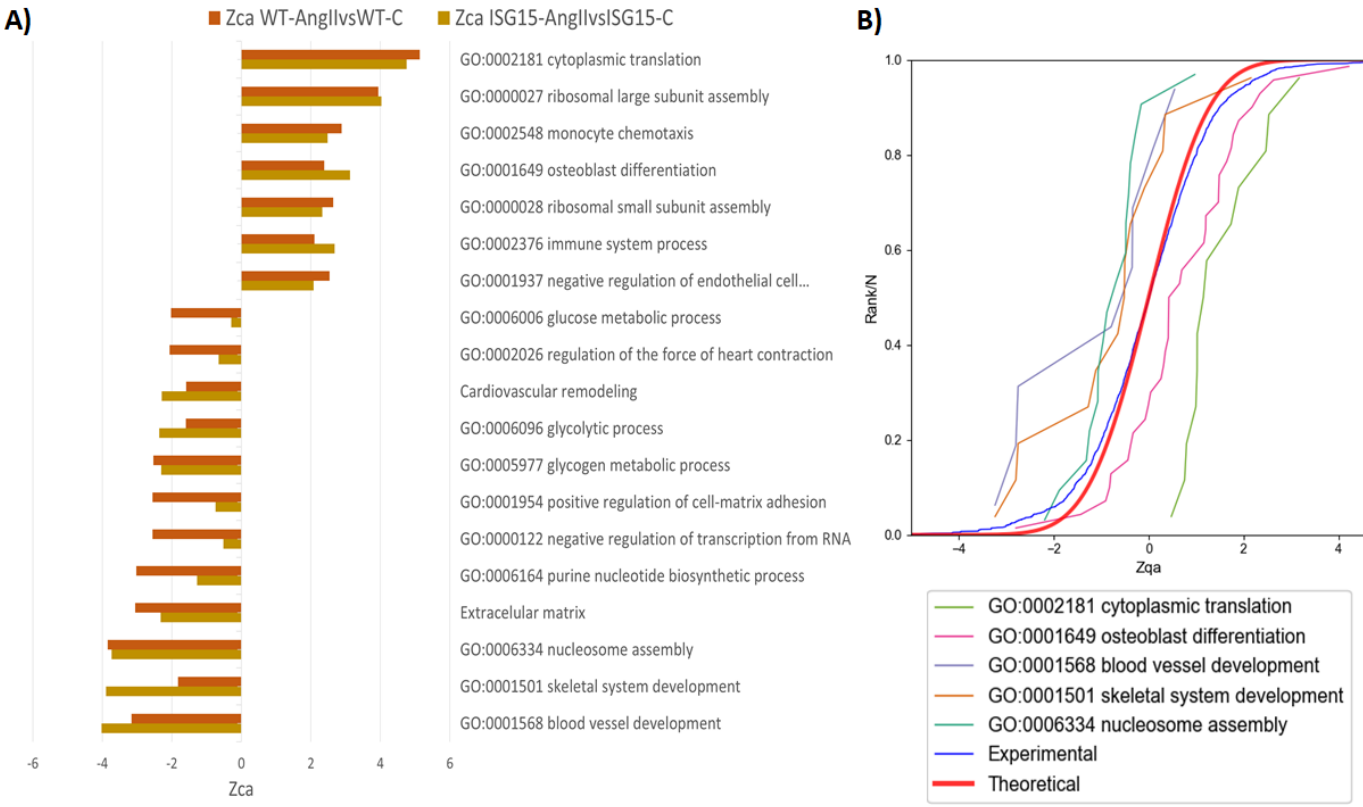


Figure 54. Functional category changes arising from coordinated protein behavior. A) Bar graph for functional categories (Zca) constructed from the "Nprot2cat_Quancat_filtered" REPORT table. B) The distributions of the standardized log2 protein ratios (Zqa) are shown for some of the functional categories that are significantly down-regulated (Left) or up-regulated (Right). The data to create the sigmoid curves are taken from the "Nprot2cat_Quancat_Quanprot_filtered" REPORT table.

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 *Importing 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 reversibly oxidized Cys 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]. These techniques introduced a differential label on Cys residues, depending on their oxidation state, producing two separate populations of reduced and oxidized Cys-containing peptides. MEF samples were incubated with diamide (treated group), or PBS (control group) and the resulting peptides were isobaricaly labeled with iTRAQ 8-plex (four biological replicates per condition). The workflow detects statistically significant abundance changes in peptides contanining modified Cys residues.

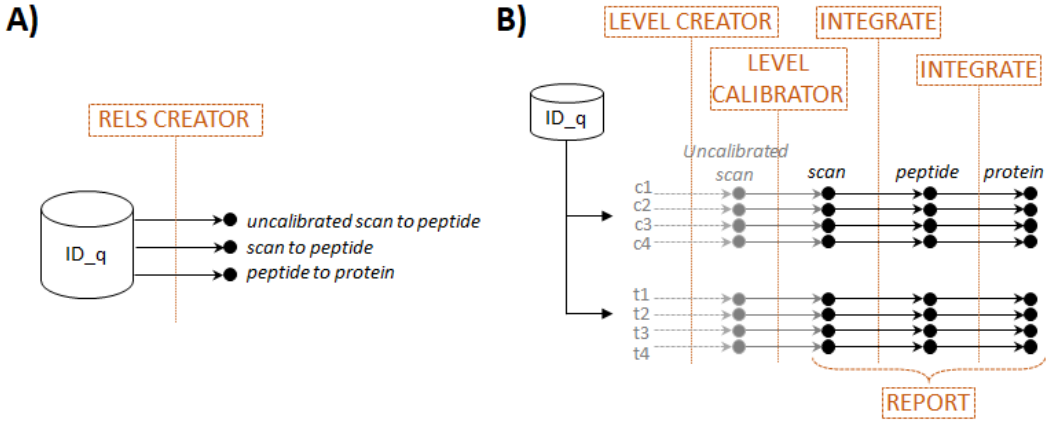


Figure 55. Scheme of workflow 3 (quantification of posttranslationally modified peptides in a labeled experiment) showing module components: RELS CREATOR (A) and LEVEL CREATOR, LEVEL CALIBRATOR, INTEGRATE, and REPORT (B).

Workflow operation

Workflow 3 comprises the basic modules LEVEL CREATOR, LEVEL CALIBRATOR, and INTEGRATE, as well as the RELS CREATOR and REPORT modules (*Figure 55*) and is very similar to workflow 2. LEVEL CREATOR was used to design the ratios and to generate the level files, sample folders and log2 ratios indicated in the corresponding task table (*Figure 56* and *Figure 57*). LEVEL CALIBRATOR was used to calibrate statistical weights (*Figure 58*, top) and INTEGRATOR to integrate from scan to peptide and from peptide to protein (*Figure 58*, bottom).

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 56. The LEVEL CREATOR task table for workflow 3.

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 57. 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).*

The only difference with workflow 2 lies in the INTEGRATE command used for the integration *peptide-to-protein*. INTEGRATE can use a modified version of the GIA algorithm for the quantitative analysis of post-translational modifications (PTM) that includes a third column containing *tags* in the relation tables, as described [7]. In this workflow the advanced option of INTEGRATE was activated to display the *Tag* column, which is used to include only the peptides which are tagged in the relation table with the text *"Not modified"* when calculating the protein averages (*Figure 58*). An example of tagged *peptide2protein* relation table is shown in *Figure 59.* Proteins are thus quantified using only peptides which are not modified in Cys. However, although these Cys peptides do not contribute to protein averages, they are assigned a *Zpq* value, which serves to evaluate whether they deviate significantly from the expected distribution of peptides around their protein averages [4]. If the deviation is statistically significant it can be concluded that there is a change in abundance of the posttranslational modification in relation to the protein it comes from. This philosophy can be extended to any other kind of PTM.

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		Not modified

Figure 58. The LEVEL CALIBRATOR (Top) and INTEGRATE (Bottom) task tables for workflow 3.

protein	peptide	Modifications
>tr|E9Q7Q3|E9Q7Q3_MOUSE Tropom	EQAEAEVASLNRR	Not modified
>tr|E9Q7Q3|E9Q7Q3_MOUSE Tropom	EQAEAEVASLNR	Not modified
>sp|Q8BTM8|FLNA_MOUSE Filamin-	SNFTVDC@SK{	Reduced-Cys peptides
>sp|P48962|ADT1_MOUSE ADP/ATP	DFLAGGIAAAVSK{	Not modified
>sp|Q8VDN2|AT1A1_MOUSE Sodium/	NLEAVETLGSTSTIC@SDK{	Reduced-Cys peptides
>sp|P20152|VIME_MOUSE Vimentin	QVQSLTC#EVDALK{	Oxidized-Cys peptides
>sp|Q9CPY7|AMPL_MOUSE Cytosol	QVIDC@QLADVNNLGK{	Reduced-Cys peptides
>sp|Q501J6|DDX17_MOUSE Probabl	GVEIC@IATPGR	Reduced-Cys peptides
>sp|Q9CZ44|NSF1C_MOUSE NSFL1 c	LGSTAPQVLNTSSPAQQAENEAK{	Not modified
>sp|B2RSH2|GNAI1_MOUSE Guanine	TTGIVETHFTFK{	Not modified

Figure 59. Excerpt from the peptide2protein relation table used to integrate peptides to proteins. Note the presence of a third column used to tag Cys-containing peptides, which will be excluded from the calculation of protein averages in the peptide-to-protein integration.

iSanXoT allows to automatically generate relation tables containing tags, which are taken from the "ID-q" table. For this end, RELS CREATOR uses a specific option (*Figure 60*). In this particular case, this option makes RELS CREATOR to look into the "ID-q" table for the column with the header *Modifications* and translate its content into the third column of the *peptide2protein* relation table. In the "ID-q" table used in this case, the peptide containing modified Cys residues were labeled as "*Reduced-Cys peptides*" and "*Oxidized-Cys peptides*" depending on the type of modification. These tags are located in the relation table by RELS CREATOR (*Figure 59* and *Figure 60*). iSanXoT allows to use any tag created by searching engines or defined by the user, with the only condition that the tag indicated in the INTEGRATE command must match the tag in the third column of the relation table (*Figure 58*, bottom).

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 60. The RELS CREATOR task table for workflow 3.

Finally, the REPORT module collects the statistical variables generated by the *peptide-to-protein* integration for all the samples (*c1, c2, c3, c4, t1, t2, t3,* and *t4*), as prompted by the asterisk (*Figure 61A*). The REPORT commands are similar to those used to generate the protein tables in workflow 1, with the difference that peptide values are tabulated instead of protein values, together with the number of scans per peptide, instead of the number of peptides per protein. The FDR at the peptide level allows to detect statistically significant changes in PTM. This REPORT also generates a second filtered peptide table containing the peptides with reduced Cys with the most extreme abundance changes. This table was used to generate a heatmap (*Figure 61B*).

A)

Sample fo	Lower level	Higher level	Reported vars	Output report	Column headers to elim	Merge with report	Ad	Filter
*	scan	peptide	n	Nscan2pep				
*	peptide	protein	Z , FDR, tags	Nscan2pep_Quanpepprot	scan	Nscan2pep		
*	peptide	protein	Z , tags	Nscan2pep_Quanpepprot_filtered	scan	Nscan2pep		(tags_peptide2protein == "Reduced-Cys peptides") & (Z_peptide2protein >= 2.5 | Z_peptide2protein <= -2.5)

B)

*Figure 61. (A) The task table for Workflow 3 is in the REPORT module. (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*). The data for the heatmap was generated from the "Nscan2pep_Quanpepprot_filtered" report table.*

Of note, the peptides included in the integration followed a standard distribution in the eight samples, as shown in blue on *Figure 62* for *t1*, *t2*, *c1*, and *c2* samples. This evidences that the error distribution at the peptide level could be accurately modeled using the GIA algorithm. In addition, the treatment produced a generalized increase in the abundance of oxidized Cys-containing peptides (orange curves), with concomitant decrease in the abundance of reduced

Cys-containing peptides (green curves). Consistently, the opposite changes were observed in the controls.

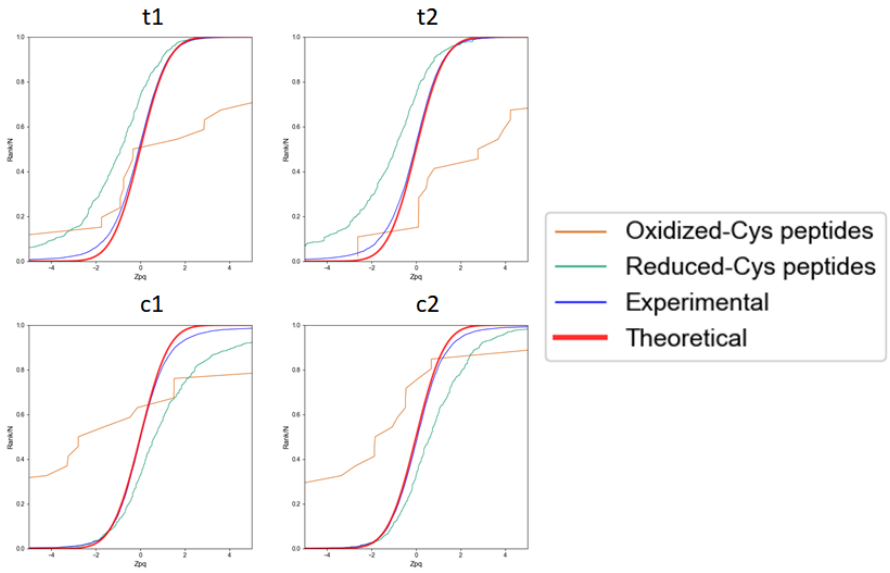


Figure 62. Distribution of the standardized variable at the peptide level (Zpq) in control MEF samples (t1, t2, c1 and c2) for all the peptides quantitated (blue) and the oxidized (orange) and reduced (green) Cys-containing peptide subpopulations. The theoretical normal distribution N(0,1) is shown in red. Positive/negative Zpq values indicate increased/decreased peptide abundance with respect to the average. These sigmoidal curves were created from the "Nscan2pep_Quanpepprot" table generated by REPORT.

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 *Importing a workflow template* Section below for detailed instructions.

Workflow 4: Label-free quantification

Experimental

This workflow was used to analyze quantitative data obtained in a multicenter study aimed at evaluating different bioinformatics tools [6]. The authors generated two hybrid proteome test 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 several software packages. In this example we show how to process the data obtained by MaxQuant [8] (see Section below to learn how data from MaxQuant and other softwares can be adapted to be used with iSanXoT).

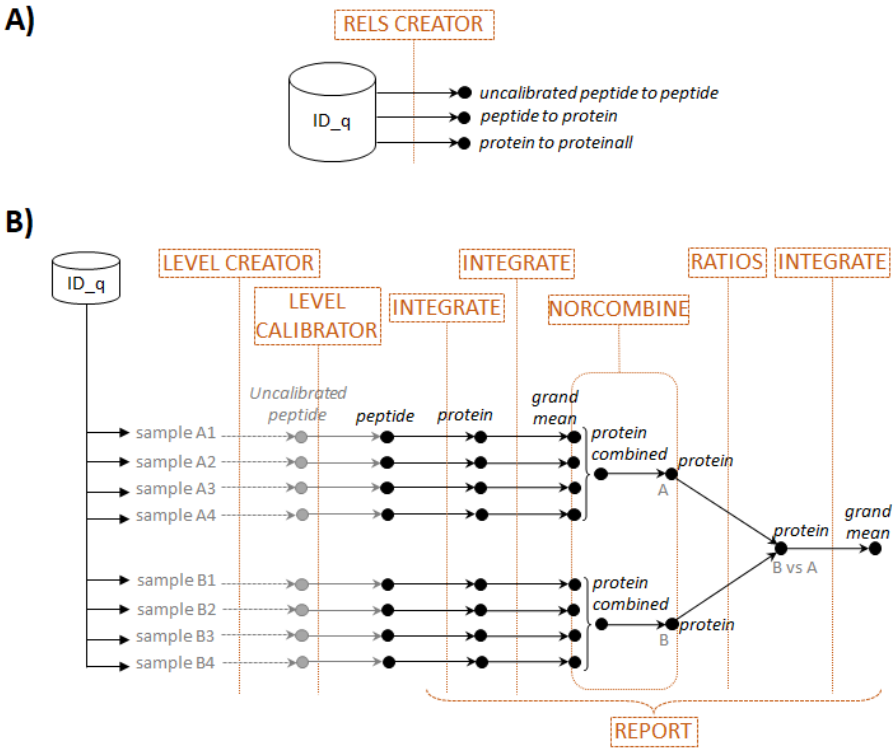


Figure 63. Scheme of workflow 4 (label-free quantification) showing module components: RELS CREATOR (A) and LEVEL CREATOR, LEVEL CALIBRATOR, INTEGRATE, NORCOMBINE, and REPORT (B).

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 63)*. The starting module, LEVEL CREATOR, generates the level files, sample folders and log2 ratios indicated in the corresponding task (*Figure 64*) based on the quantitative data at the peptide level obtained with MaxQuant for replicate *A*- and *B*-type samples. In this example we used as denominator of the log2ratio the average of peptide intensities across all the samples. In this case, however, the averages of the four *A*-type and the four *B*-type samples are first

calculated separately (as indicated by the square brackets), then the average of the two averaged values is calculated (as indicated by the comma). This ensures that no log2-ratio is calculated when the four values are missing in either the *A* or the *B* sample group. This module generates uncalibrated files at the peptide level (*u_peptide*) (*Figure 65*).

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 64. The LEVEL CREATOR task table for workflow 4.

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 65. 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 uncalibrated statistical weights (right column).

The *u_peptide* level files are then calibrated with the LEVEL CALIBRATOR module by performing an integration to the protein level (*Figure 66, Top*), generating calibrated *peptide* level files. The *peptide-to-protein* and *protein-to-proteinall* integrations are then performed by the INTEGRATE module according to the module task table (*Figure 66*, Bottom).

Note that in this example the advanced option of INTEGRATE was activated to use the Tag column, so that only the proteins containing the *Homo sapiens* tag are used in the protein-to-proteinall integrations (*Figure 66, Bottom*). Restricting the integration to human proteins serves for two purposes: a) the normalization is done by the grand mean of the human proteins and is not affected by the presence of yeast or *E. coli* proteins. b) only human proteins are used to estimate the variance of the *protein-to-protein* all integration, avoiding the effect of yeast and E. coli proteins, whose ratios have a large deviation from the mean. Note that this procedure does not eliminate yeast or *E. coli* proteins from the normalized files that are later used by the NORCOMBINE module (see below).

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

As in workflow 3, the *protein-to-proteinall* relation table must contain a third column tagging the species from which each protein comes from (*Figure 67*). Note that the tag indicating the human proteins matches the tag indicated in INTEGRATE (*Figure 66, Bottom*).

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 67. 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.

The *protein-to-proteinall* relation table is automatically created by the RELS CREATOR module, which takes this information from a file (HumanSaccEcoliPME12_divide_by_species.tsv) created by the user. The file contains the relationship between protein identifiers (*Protein_Id* column header) and the species (*Species* column header) they come from.

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	{PATH}/HumanSaccEcoliPME12_divide_by_species.tsv

Figure 68. The RELS CREATOR task table for workflow 4.

Next, the normalized data at the *protein* level from the four replicates from each sample are combined into samples *A* and *B*, respectively, using the NORCOMBINE basic module (*Figure 69*, Top). To compare these two samples, new log2 ratios and statistical weights are calculated

using the RATIOS basic module (*Figure 69*, Bottom). Finally, a *protein-to-proteinall* integration is carried out for the newly-generated *B_vs_A* sample by the module INTEGRATE (*Figure 66*, Bottom), using again the *Homo sapiens* tag.

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

The REPORT module used in this workflow tabulates the data at protein level together with the number of peptides per protein, as in previous workflows (*Figure 70*). In this example, besides *Z* and *FDR*, the table includes the log2ratios of proteins from all the samples (*Xinf* from the *protein-to-proteinall* integration, to which we also refer as *Xq*), the grand mean (*Xsup* from the *protein-to-proteinall integration,* or *Xa*), the statistical weights (*Vinf* or *Vq*) and the species each protein belongs to (*tags*). The grand mean, which is used to normalize log2-ratios, and the statistical weights can be used to construct plots such as in *Figure 71* (see below).

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

Figure 70. The REPORT module task table for workflow 4.

It is noteworthy that variance modelling, normalization, standardization and statistical weighting, according to the GIA algorithm, are performed automatically, without data filtering, pre-processing or missing value imputation [6], even in a situation where numerous proteins have highly imbalanced data. Moreover, the sigmoid plots automatically generated in each one of the integrations performed by INTEGRATE clearly demonstrate that the GIA algorithm accurately predicts the distribution of peptide quantifications around their proteins (*Figure 71A*) and of protein quantifications around the grand mean (*Figure 71B*). These results demonstrate that this statistical model is very suitable for the analysis of label-free data.

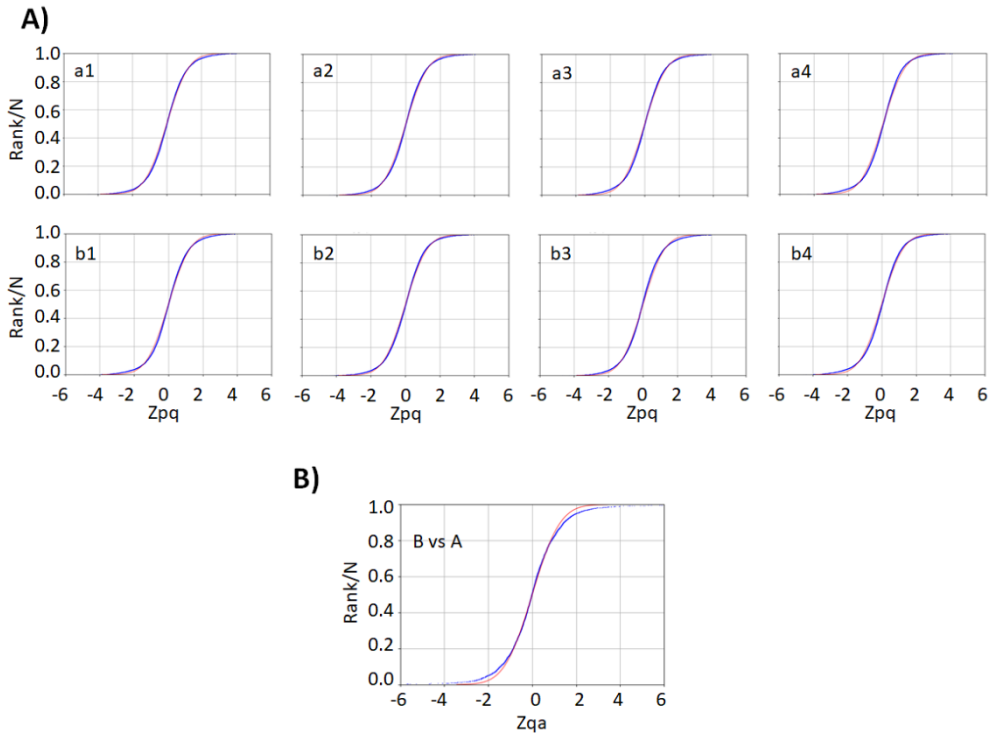


Figure 71. Distribution of the standardized variable at the peptide (Zpq) and protein (Zqa) levels for label-free data analyzed with iSanXoT. A) Zpq distribution for the eight individual A-type and B-type samples. B) Zqa distribution for the B-type vs A-type comparison. Red: null hypothesis (standard distribution); blue: experimental data.

The combined statistics *B_vs_A* also shows how human, yeast and bacterial proteins distribute around the expected 0, 1, and -2 log2- values (corresponding to 1-, 2- and 0.25-fold changes) (*Figure 72)*, and how protein quantifications with higher statistical weights are more accurate. This plot also confirms how iSanXoT provides highly accurate quantitative results in a fully automated fashion.

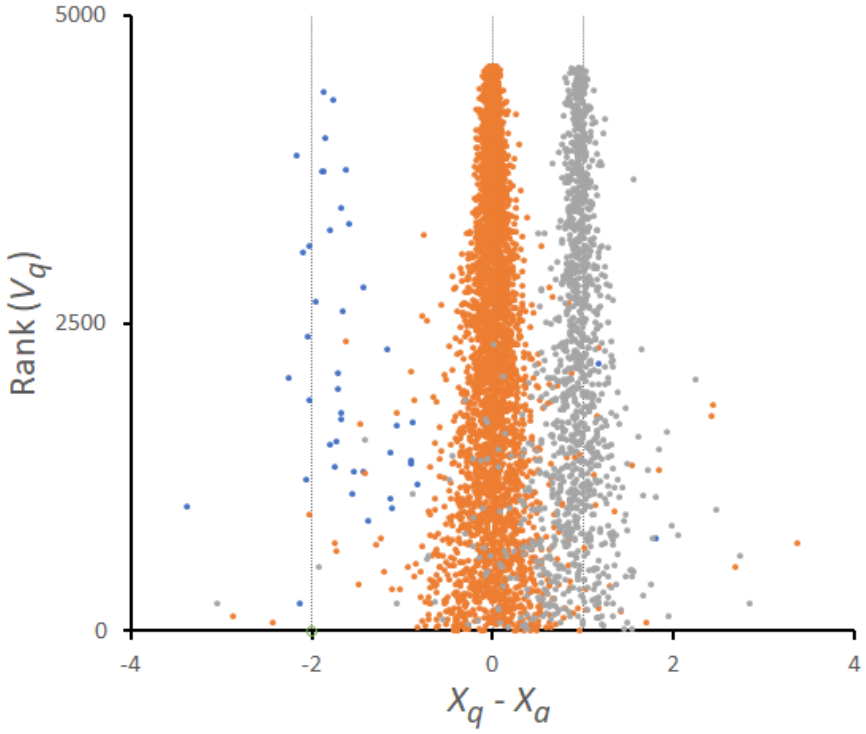


Figure 72. Quantification of human (orange), yeast (grey) and bacterial (blue) proteins according to the combined statistics B_vs_A. Shown are log2-ratios normalized by the grand mean (Xq – Xa or Xinf - Xsup). This plot was generated from the table "Npep2prot_Quanprot".

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 *Importing a workflow template* Section below for detailed instructions.

Importing a workflow template

In this section we will provide instructions to execute the workflow examples and to import workflows that were previously created with iSanXoT to be reused in other projects. We will use as example the first workflow described in the previous section.

Start by downloading workflow 1 template and input files from the iSanXoT wiki (https://github.com/CNIC-Proteomics/iSanXoT/wiki/studies/workflows/WSPP-SBT.zip). Then extract the files included in the compressed archive to create a folder named WSPP-SBT. Check that the WSPP-SBT folder has been created in your file system. Then proceed as follows:

- Open the iSanXoT application by double-clicking the application icon (*Figure 73*).

Figure 73. The iSanXoT startup message.

- Choose *New Project* from the *Project* menu (*Figure 74*).

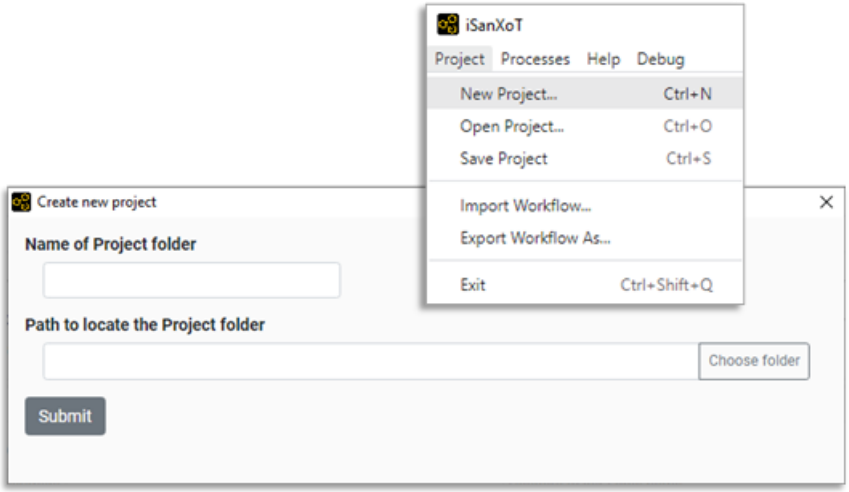


Figure 74. Create New Project.

- Provide a name of your choice for the project folder and indicate a path to locate this folder, then click the *Submit* button (*Figure 74*).
- Choose *Import Workflow* from the *Project* menu (*Figure 75)* and select the folder WSPP-SBT created before (or any other iSanXoT project folder from which you want to import the workflow).

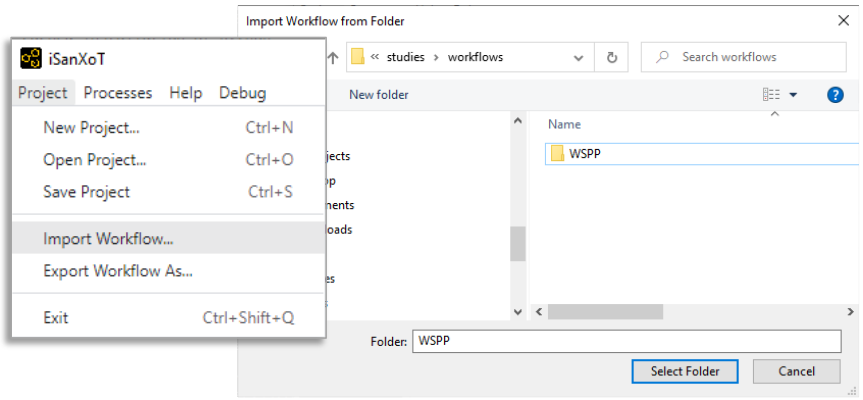


Figure 75. Importing a preexisting iSanXoT workflow to the newly-created project.

- Inspect the WSPP-SBT task table (in the Compound modules tab), the RELS CREATOR task table (in the Relation tables tab) and the REPORT task table (in the Reports tab) to check that the tables indicated in Fig. S2, S3 and S5 have been correctly loaded. Note that if a different template is imported, only the corresponding task tables will be loaded.

- Now click on *Choose identification file* and select "ID-q.tsv" in the WSPP-SBT folder (*Figure 76)*. Or, alternatively, select the desired identification/quantification table with which this workflow is to be executed. Section 3 below shows how to prepare the "ID-q" file based on the output from a variety of proteomics pipelines. Bear in mind that the tasks defined in the LEVEL CREATOR and RELS CREATOR modules have to match the samples and column names from the specific "ID-q" file used.

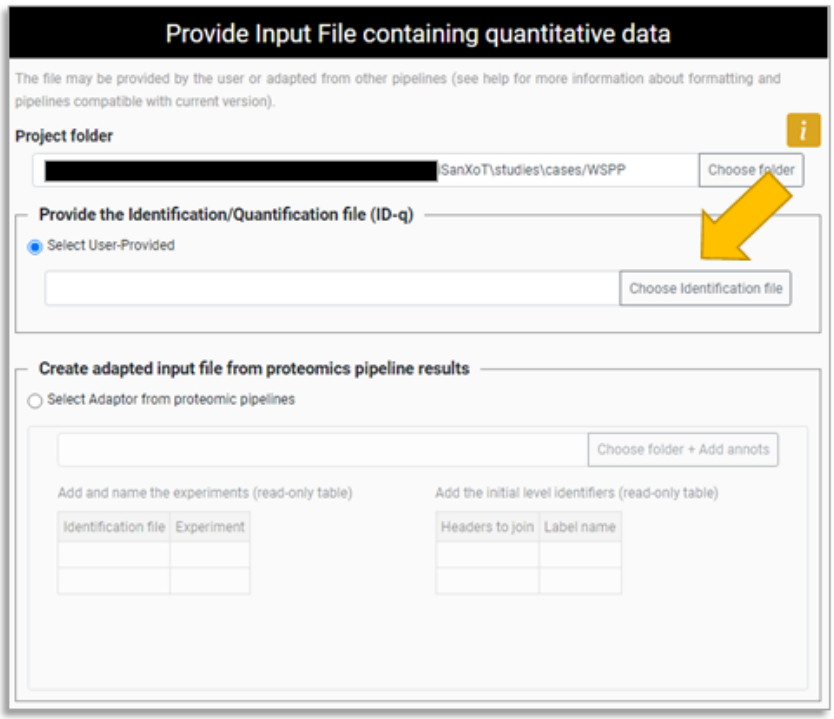


Figure 76. Choosing the identification/quantification (ID-q) file for the newly-created project.

- Select *Save Project* from the *Project* menu to save the changes or directly press the *Save*

and Run button to save and execute the current workflow.

Creating the identification/quantification file from proteomics pipelines

iSanXoT requires an identification/quantification file in tsv format (*ID-q.tsv*) containing at least the quantified features together with their quantitative values. Any tsv table may be used as an *ID-q* file, provided that quantitative values are arranged in a table with column headers, so that features (e.g. PSMs or peptides) are arranged in rows and their quantitative values in columns, where every column pertains to a different sample. The column headers of the *ID-q* are used by iSanXoT to extract the necessary information.

In addition, when the *ID-q* file contains features quantitated in more than one experiment (e.g. different samples labelled with the same TMT-18plex tags), an additional column with the header *Experiment* must be included indicating the experiment ascription of the features.

Finally, iSanXoT also needs information to create the relation files required to integrate the quantified features into higher levels. This information is usually also present in the *ID-q* file. For instance, iSanXoT can use the columns containing the scan and the peptide identifiers to construct the *scan2peptide* relation table.

The majority of proteomics software tools generate tables that can be easily used for this purpose. In this Section we shall describe how to prepare the *ID-q* file based on the output from the three most popular proteomics pipelines (*Table S1*).

Table S1. Output data from proteomics pipelines to be included in the ID-q.tsv *file.*

Proteomics pipeline	Experiment type	Output file name / suffix	Level name in the output file[1]			Quantitative data
			Scan	Peptide	Protein	
Proteome Discoverer (version 2.5)	Label-free	*_PeptideGroups.txt*		*Sequence + Modifications*	*Master Protein Accessions*	*Abundance: FX: Sample Type*
	Isotopically labelled	*_PSMs.txt*	*Spectrum File + First Scan*[2]	*Sequence + Modifications*	*Master Protein Accessions*	*Abundance: Quan Channel*
MaxQuant (version 1.6.5.0)	Label-free	*modificationSpecificPeptides.txt*		*Sequence + Modifications*	*Proteins*	*Intensity Experiment*
	Isotopically labelled	*msmsScans.txt*	*Raw file + Scan number*	*Modified Sequence*	*Proteins*	*Reporter intensity n*
Fragpipe (version 1.8.1)	Label-free	*combined_modified_peptide.tsv*		*Modified Sequence*	*Protein ID*	*Experiment Intensity*
	Isotopically labelled	*psm.tsv*	*Spectrum + Spectrum File*[3]	*Modified Peptide*	*Protein ID*	*Channel*

[1]If features quantitated in multiple experiments (e.g. different samples labelled with the same TMT-18plex tags) are to be considered, an additional column with the header *Experiment* must be included indicating the experiment ascription of the features.

[2]Make sure *Max. Number of Peptides Reported* = 1 was selected in the *Input Data* section of the Proteome Discoverer *Processing* node used.

[3]Make sure *Report top N* = 1 was selected in the *Advanced Output Options* of the FragPipe *MSFragger* module.

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

In the case of Proteome Discoverer version 2.5 [9], the way that quantitative data are adapted for use with iSanXoT depends on whether they originate from label-free or labelled experiments:

Label-free experiments

In this case, quantitative data at the peptide level can be adapted for use with iSanXot from the *_PeptideGroups.txt* files obtained when the *Processing* workflow node *Minora Feature Detector* of Proteome Discoverer is used. The following column headers of the *_PeptideGroups.txt* files must be considered for preparing the *ID-q* file:

- *Sequence*: Amino acid sequence of the identified peptide;
- *Modifications*: Chemical or posttranslational modifications to the *Sequence* above;
- *Master Protein Accessions*: Accession code(s) for the protein(s) to which the peptide *Sequence* is ascribed;
- *Abundance: FX: Sample Type:* Peptide intensity in the RAW file identified with *FX* and tagged as *Sample Type* in the Proteome Discoverer *Input Files* tab.

The *peptide* level required for the *peptide to protein* integration with iSanXoT can be obtained by merging the *Sequence* and *Modifications fields* (see Section *Adapting the results from proteomics pipelines for iSanXoT* below).

Labelled experiments

For labelled experiments (e.g. TMT- or iTRAQ-based), quantitative data at the scan level can be adapted for use with iSanXot from the *_PSMs.txt* files generated when the *Processing* workflow node *Reporter Ions Quantifier* of Proteome Discoverer is used. The following column headers of the *_PSMs.txt* files must be considered for preparing the *ID-q* file:

- *Spectrum File*: Name of the RAW file where the PSM was identified;
- *First Scan*: Spectrum (scan) number of the PSM in the RAW file;
- *Sequence*: Amino acid sequence of the identified peptide;
- *Modifications*: Chemical or posttranslational modifications to the *Sequence* above;
- *Master Protein Accessions*: Accession code(s) for the protein(s) to which the peptide *Sequence* is ascribed;
- *Abundance: Quan Channel*: Intensity of the reporter ion tagged as *Quan Channel* in the Proteome Discoverer *Samples* tab.

For the *scan to peptide* integration with iSanXoT, the *scan* level can be obtained by merging the *Spectrum File* and *First Scan* fields, and the *peptide* level by merging the *Sequence* and *Modifications* fields (see Section *Adapting the results from proteomics pipelines for iSanXoT* below; make sure *Max. Number of Peptides Reported* = 1 was selected in the *Input Data* section of the Proteome Discoverer *Processing* node used).

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 quantifications at the peptide level required to prepare the *ID-q* file can be found in the *modificationSpecificPeptides.txt* file, which is stored in the *"…combined/txt*" folder. The following column headers of the *modificationSpecificPeptides.txt* file must be considered for preparing the *ID-q* file:

- *Sequence*: Amino acid sequence of the identified peptide;
- *Modifications*: Chemical or posttranslational modifications to the *Sequence* above;
- *Proteins*: Identifier(s) of the protein(s) to which the peptide *Sequence* is ascribed;
- *Intensity Experiment*: Summed up extracted ion current of all isotopic clusters associated with the peptide Sequence identified across the raw files included in the *Experiment* as specified by the user in the MaxQuant *Raw data* tab.

The *peptide* level required for the *peptide* to protein integration with iSanXoT can be obtained by merging *Sequence* and *Modifications* fields (see Section *Adapting the results from proteomics pipelines for iSanXoT* below).

Labelled experiments

When dealing with labelled experiments (e.g. iTRAQ- or TMT-based), the necessary quantitative data at the scan level can be found in the *msmsScans.txt* file, which is stored in the *"…combined/txt*" folder. The following column headers of the *modificationSpecificPeptides.txt* file must be considered for preparing the *ID-q* file:

- *Raw file*: Name of the RAW file where the PSM was identified;
- *Scan number*: Spectrum (scan) number of the PSM in the RAW file;
- *Modified Sequence*: Amino acid sequence of the identified peptide including chemical or posttranslational modifications. 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 *n* as specified by the user in the MaxQuant *Group-specific parameters* tab.

For the *scan to peptide* integration with iSanXoT, the *scan* level can be obtained by merging the *Raw File* and *Scan number* fields (see Section *Adapting the results from proteomics pipelines for iSanXoT* below).

Preparing the *ID-q* file from FragPipe output

The way that quantitative data from Fragpipe version 1.8.1 [10] are adapted for use with iSanXoT depends on whether they originate from label-free or labelled experiments:

Label-free experiments

FragPipe *Quant (MS1)* module stores the quantifications at the peptide level necessary to prepare the *ID-q* file in a *combined_modified_peptide.tsv* file. The following column headers of the *modificationSpecificPeptides.txt* file must be considered for preparing the *ID-q* file:

- *Modified Sequence*: Amino acid sequence of the identified peptide;

- *Protein ID*: Identifier of the protein to which the *Modified Sequence* peptide is ascribed;
- *Experiment Intensity*: Summed up intensity of the *Modified Sequence* peptide in the RAW files included in the *Experiment* as specified by the user in the FragPipe *Workflow* tab.

Labelled experiments

Fragpipe *Quant (Isobaric)* module generates a *psm.tsv* output file that contains the quantitative data at the scan level obtained from labelled experiments. The following column headers of the *psm.txt* file must be considered for preparing the *ID-q* file:

- *Spectrum*: Spectrum (scan) identifier of the PSM in the XML file;
- *Spectrum File*: Name of the XML file where the PSM was identified;
- *Modified Peptide*: Amino acid sequence of the identified peptide including chemical or posttranslational modifications;
- *Protein ID:* Identifier of the protein to which the *Modified Peptide* is ascribed;
- *Channel*: Intensity of the reporter ion *Channel* as specified by the user in the FragPipe TMT-Integrator table of the *Quant (Isobaric)* module.

The *scan* level required for the later *scan to peptide* integration with iSanXoT can be obtained by merging the *Spectrum* and *Spectrum File* fields (see Section *Adapting the results from proteomics pipelines for iSanXoT* below; make sure *Report top N = 1* was selected in the *Advanced Output Options* of the FragPipe *MSFragger* module).

Adapting the results from proteomics pipelines for iSanXoT

iSanXoT requires an identification/quantification tab-separated values file (*ID-q.tsv*) containing at least the identified features together with their quantitative values (an experiment identifier is required if two or more experiments are included). Users can either compose this *ID-q* file manually (see the previous Section 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.

- Run the iSanXoT application and create a new project or open an existing project. A new window will appear asking for the ID-q file (*Figure 77*).

- If you already have a suitable *ID-q* file, click the *Select User-Provided* option and then *Choose identification file* to select the file (*Figure 77*).

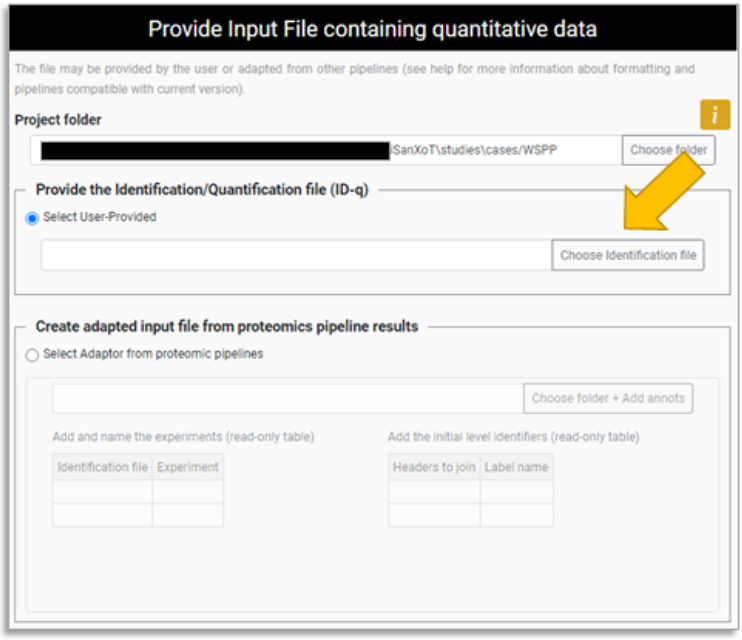


Figure 77. Selecting an ID-q file in the Input Adaptor main window.

- If you do not have an ID-q file, click *Select Adaptor from proteomics pipelines*. This option will launch the iSanXoT adapter to import your quantitative data. The adapter has been tested with recent versions of MaxQuant, Trans-Proteomic Pipeline, FragPipe, and Proteome Discoverer. Click on *Choose folder + Add anots* to select the folder containing your quantitative data (*Figure 78*). A three-panel window will pop-up (*Figure 79*).

Figure 78. Having the iSanXoT Input Adaptor prepare the ID-q file.

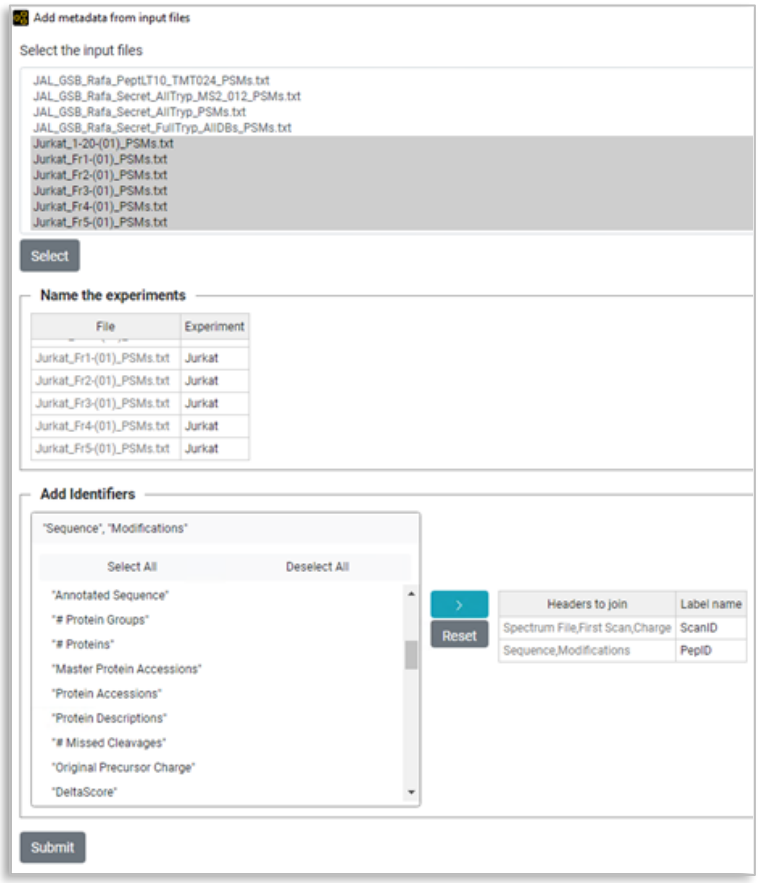


Figure 79. Adapting results from a proteomics pipeline. In the top panel several output files from Proteome Discoverer have been selected. These PSMs.txt files, which contain identification/quantification data, have been assigned an experiment name (Jurkat) in the middle panel. The bottom panel has been 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 the adapter. 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] Bagwan N, Bonzon-Kulichenko E, Calvo E, Lechuga-Vieco AV, Michalakopoulos S, Trevisan-Herraz M, et al. Comprehensive Quantification of the Modified Proteome Reveals Oxidative Heart Damage in Mitochondrial Heteroplasmy. Cell Rep. 2018;23(12):3685-97 e4.
[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] 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.