iSanXoT -0.4.1

Introduction

Get Started

TALK ABOUT THE INTERFACE,

EXPLAIN THE MENU AND THE RELATION TO THE TABS

Modules

The iSanXoT desktop application houses several modules based on the SanXoT software package [1]. The execution of each module is done and is described by means of a Task-Table (TT). These TT represent the parameters that are needed to the setting up and the execution of module.

There are three types of modules:

- Basic modules: These modules are inherently based on the SanXoT packages [1]. They represent the minimal expression of integration in SanXoT.
- Report modules: These modules are used to
- Complete modules:

Basic modules

RELS CREATOR

This module parse data from a given tabular file, extract several columns creating a file with tab separated table, Relation Table (RT). These RT created, are used by the integrations in the rest modules. For this reason, the name of RT indicated by the parameter/column of its task-table ("Relation Table to be created"), has to coincide with the "lower-level" and the "higher-level" of the integration; e.g. whether an integration is between the "peptide" to "protein" levels, the RT applied has to be called "peptide2protein".

Simple parameters

The parameters/columns of this module are:

- Forced execution: checkbox that determines to force the execution or not.
- Relation Table to be created: indicates the file name of relation table. This name must be formed with the lower level to higher level for one integration.
- Column name of Lower level: column name from the given tabular file that will use for the lower level in the resulted RT.
- Column name of Higher level: column name from the given tabular file that will use for the higher level in the resulted RT.
- Column name of 3rd column: column name from the given tabular file that will use for the third column in the resulted RT.
- Table from which RT is extracted: file name with path which the levels are extracted. This file has to be in tabular format. In this parameter the keywords can be used. For more details read "Keywords for the Task-Tables" in the "Special parameters" section.

F. exec.	Relation Table to be created	Column name of Lower level	Column name of Higher level	Column name of 3rd column	Table from which RT is extracted
☐	uscan2peptide	Scan_Id	Peptide		__IDQFIL__
☐	scan2peptide	Scan_Id	Peptide		__IDQFIL__
☐	peptide2protein	Peptide	Protein		__IDQFIL__
☐	peptide2peptideall	Peptide	[1]		__IDQFIL__
☐	protein2proteinall	Protein	[1]		__IDQFIL__
☐	protein2description	Protein	Protein_Description		__IDQFIL__
☐	protein2category	Protein	cat_*		S:\U_Proteomica\UNIDAD\iSanXoT_DBs\202105\human_202105.pid2cat.tsv
☐	category2categoryall	cat_*	[1]		S:\U_Proteomica\UNIDAD\iSanXoT_DBs\202105\human_202105.pid2cat.tsv
☐	protein2gene	Protein	Gene		S:\U_Proteomica\UNIDAD\iSanXoT_DBs\202105\human_202105.categories.tsv

Figure 1. Task-Table example of RELS CREATOR module.

LEVEL CREATOR

The LEVEL CREATOR module creates a level that it the quantitative data. This data file is text file that contains three columns: identifier (a text string that is used to identify the element), quantitative value (X: 2-base logarithm of the ratio of the two quantitative measurements) and prior weight (V: a parameter that measures the accuracy of the quantitative value before performing the integration).

This module takes the preliminary data from a Quantification table (ID-q file) that is provided by the user applying the “Main Input” adaptor, or it is created by one of adaptors. For more details, read the “Adaptors” section.

Simple parameters

The parameters/columns of this module are:

- Forced execution: checkbox that determines to force the execution or not.
- Experiment: is the name of experiment. This value has to be in a column called ‘Experiment’ within the Quantification data (ID-q file). Only the data with the value specified in 'Experiment' column will be extracted.
- Ratio numerator column: specifies which sample quantification values make up the numerator for the calculation of log2-ratio values.
- Ratio denominator column(s): specifies which sample quantification make up the denominator for the calculation of log2-ratio values.
- Level to be created: designates the level name.
- Output Sample folder: indicates the name of the folder where the data file of level will be saved.

F. exec.	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

Figure 2. Task-Table example of LEVEL CREATOR module.

LEVEL CALIBRATOR

This module calibrates the V values of a level by performing the specified integration. This calibration is done using the "klibrate" program developed in the SanXoT software package [1].

To perform the calibration two parameters have to be calculated: the k (weight constant), and the variance. They are calculated iteratively using the Levenberg-Marquardt algorithm. More detailed information may be found in "klibrate" integration from SanXoT software package [1].

The calibrated level contains the same information as the original level from LEVEL CREATOR module, but changing the values of the third column (containing the weights) to adapt the information to the calibrated weights that can be used as input in the INTEGRATE module.

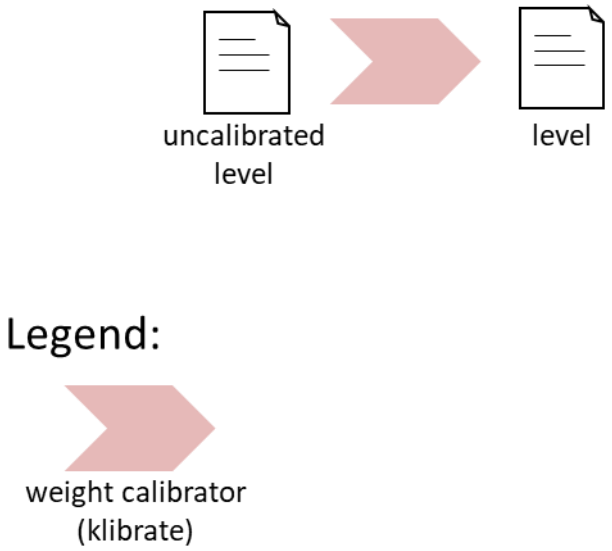


Figure 3. Schema of LEVEL CALIBRATOR module.

Simple parameters

The parameters/columns of this module are:

- Forced execution: checkbox that determines to force the execution or not.
- Sample folder(s): indicates the names of the folders where the uncalibrated data is located. These data files have been created by LEVEL CREATOR module.
- Lower level for integration: indicates the name of lower-level for the integration during the calibration.
- Higher level for integration: indicates the name of higher-level for the integration during the calibration.
- Name of calibrated level: designates the level name.
- Output Sample folder: indicates the name of the folder where the data file of level will be saved.

F. exec.	Sample folder(s)	Lower level for integration	Higher level for integration	Name of calibrated level	Output Sample folder
☐	Junkat_WT/WT_1	u_scan	peptide	scan	Junkat_WT/WT_1
☐	Junkat_WT/WT_2	u_scan	peptide	scan	Junkat_WT/WT_2
☐	Junkat_WT/WT_3	u_scan	peptide	scan	Junkat_WT/WT_3
☐	Junkat_WT/WT_4	u_scan	peptide	scan	Junkat_WT/WT_4
☐	Junkat_KO/KO_1	u_scan	peptide	scan	Junkat_KO/KO_1
☐	Junkat_KO/KO_2	u_scan	peptide	scan	Junkat_KO/KO_2
☐	Junkat_KO/KO_3	u_scan	peptide	scan	Junkat_KO/KO_3
☐	Junkat_KO/KO_4	u_scan	peptide	scan	Junkat_KO/KO_4

Figure 4. Task-Table of LEVEL CALIBRATOR module.

INTEGRATE

The INTEGRATE module performs WSPP-statistics by applying iteratively the Generic Integration Algorithm (GIA) [2] on the data. Each GIA integration performs an independent statistic and its output may be used as input for the following integration step.

This module is applied to integrate lower-level data to higher-level data. For example, to integrate from peptide-level data to protein-level, or to integrate from protein-level data to gene-level.

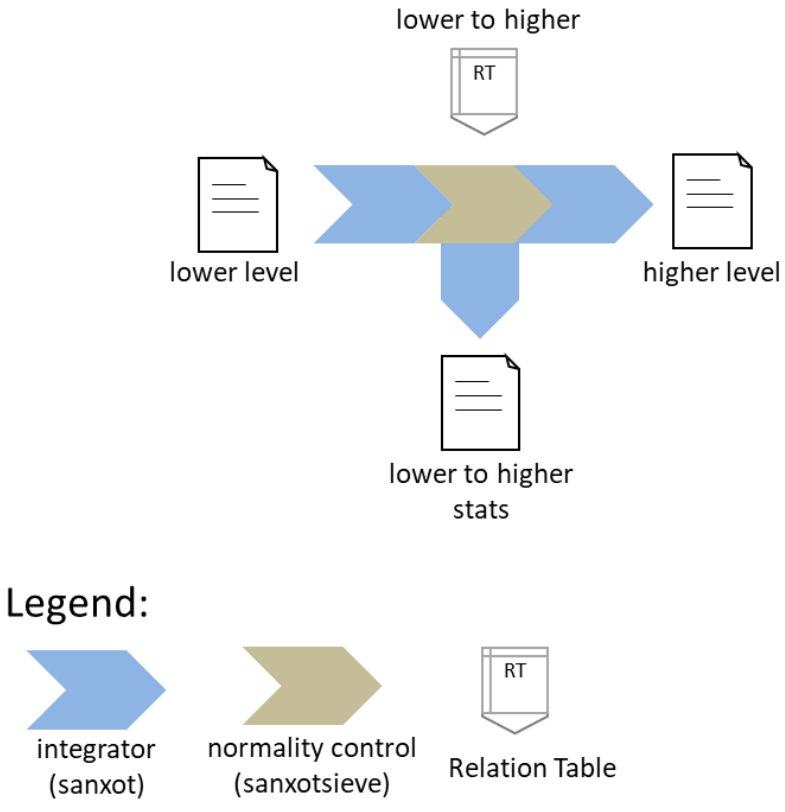


Figure 5. Schema of INTEGRATE module*. Integration from any lower-level to any higher-level, using the Generic Integration Algorithm (GIA) which is composed by the programs: "sanxot" and "sanxotsieve".*

More in detail

Each INTEGRATE needs as input tab-separated text tables:

1. Data file, that contains the quantitative data of the lower level. This file contains three columns: identifier (a text string that is used to identify the element), quantitative value (2-base logarithm of the ratio of the two quantitative measurements) and prior weight (a parameter that measures the accuracy of the quantitative value before performing the integration).
2. Relation table (RT), which establishes the correspondence of the identifiers in the lower level with those in the higher level. This file contains two columns, the first one has the identifiers of the higher level and the second one the identifiers of the lower level.

The INTEGRATE generates an output data file containing the quantitative data of the higher level. This file can be used as input for the others modules.

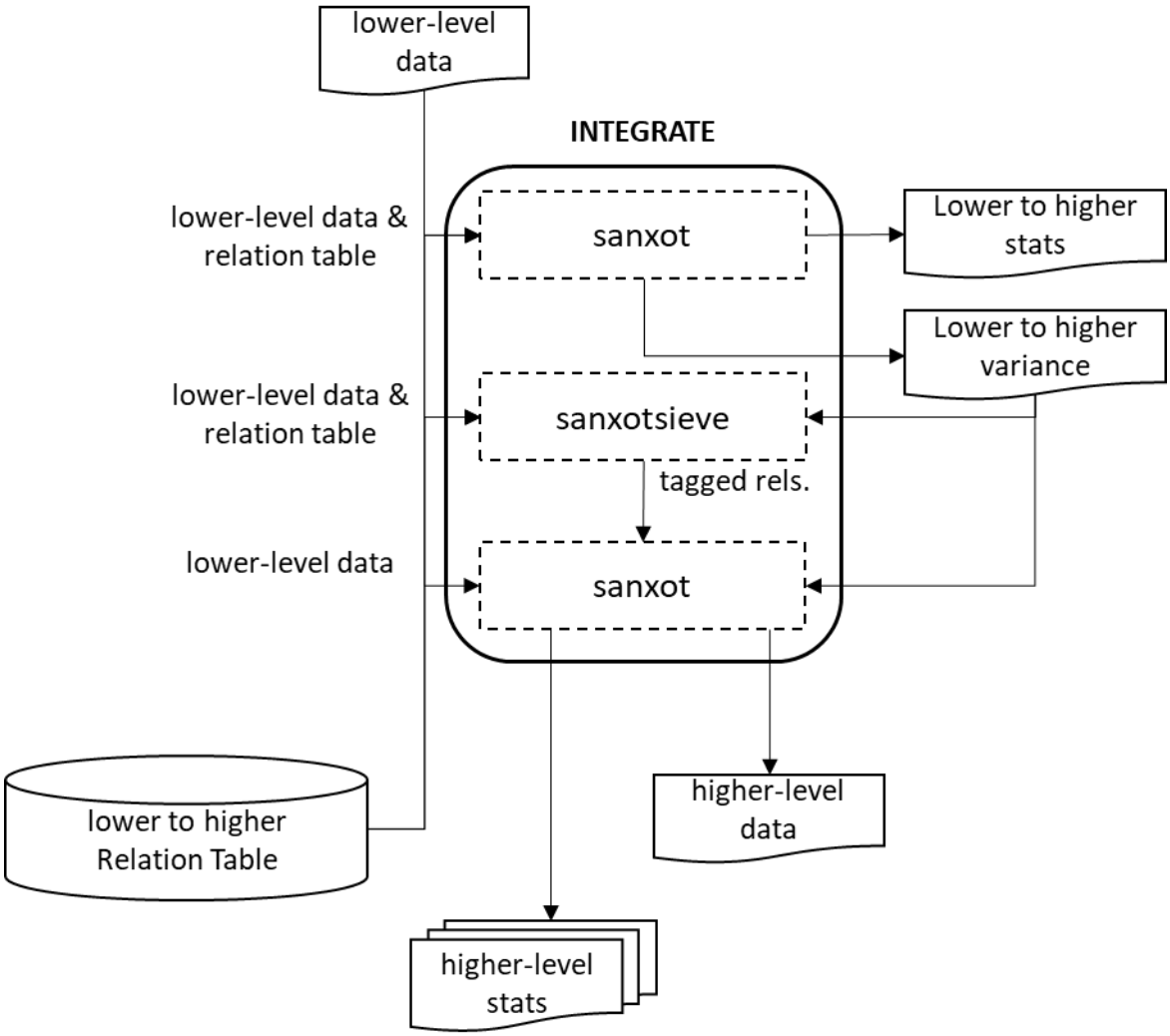


Figure 6. Flowchart of INTEGRATE module. *A first integration is done with "sanxot" that calculates the variance; "sanxotsieve" removes the outliers tagging them in a new relation table; a second integration is done with "sanxot" using the fixed variance calculated in the first step and removing the outliers tagged in the relations table.*

Each integration takes the input data file from the lower-level and the RT, calculates the general variance of the integration (using a robust iterative method) and generates as output a data file containing the integrated quantifications of the higher-level. The integration is done using the "sanxot" program.

In addition to the quantitative data, each integration may generate several additional files which contain information about the integration. More detailed information may be found in exploring SanXoT program [1].

One of the advantages of this integration is that it provides a straightforward method to eliminate outliers [3] in each integration step. This is done by calculating standardized log2-ratios (z-values) allowing to estimate the probability for each element of the lower level is a significant outlier of the z distribution (e.g., N(0,1)) and also to obtain the associated FDR. The most extreme outliers may be removed sequentially and the integration repeated, until all outliers below a user-defined FDR-level are removed.

Outlier removal is performed by the "sanxotsieve" program by tagging the outlier elements in a new relation table. Then, a second integration is done by "sanxot" program using the fixed variance calculated in the first integration ("sanxot" program) and discarding the outliers tagged in the relations table.

Simple parameters

The parameters/columns of this module are:

- Forced execution: checkbox that determines to force the execution or not.
- Sample folder(s): indicates the names of the folders where the lower-level data is located.
- Lower level: indicates the name of lower-level. The module will use the data file with the same name and it contains the three columns with the identifier, quantitative value and prior weight.
- Higher level: indicates the name of higher-level to integrate.

	F. exec.	Sample folder(s)	Lower level	Higher level
1	☐	Junkat_WT/WT_1	peptide	protein
2	☐	Junkat_WT/WT_1	protein	category
3	☐	Junkat_WT/WT_1	protein	proteinall
4	☐	Junkat_WT/WT_1	category	categoryall
5	☐			

Figure 7. Task-Table example for INTEGRATE module.

The Task-table is the way to execute the module. Each row represents one or more executions.

This module performs an integration from an existing lower-level to a higher-level using a previously created Relation Table whose name matches the higher and lower levels (e.g. if the user wants to integrate from peptide to protein, the module will use the Relation Table called scan2peptide.

Advanced parameters

This module accepts advanced parameters:

- Tag: label that below to accept or discard elements is the calculation of variance. By default, the “outliers” are discarded.
- FDR: limit of False Discovery Rate different than 0.01 (1%). If FDR = 0, then the “outliers” are not discarded.
- Var(x): force a variance. By default, the variance calculated by first “sanxot” is applied.
- More params: allows to add more parameters to the programs inside the program. For more detail read “More params” in “Special Parameters” section.

F. exec.	Sample folder(s)	Lower level	Higher level	Output Sample folder	Tag	FDR	Var(x)	More params
☐	Junkat_WT/WT_1	peptide	protein					
☐	Junkat_WT/WT_1	protein	category					
☐	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	protein	proteinall					
☐	Junkat_WT/WT_2	category	categoryall					

Figure 8. Task-Table example with advanced parameters for INTEGRATE module.

NORCOMBINE

NORCOMBINE is a module to integrate technical or biological replicates. For example, there are multiple proteins obtained from integrations separately for each replicate and then the protein-level data are integrated to obtain protein averages (grouped-level data).

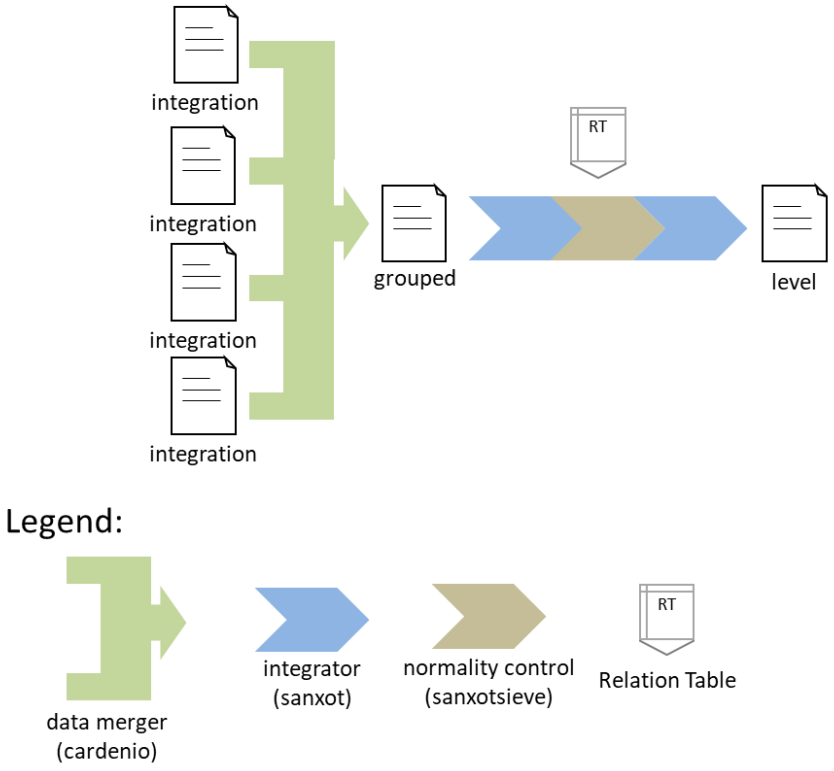


Figure 9. Schema of NORCOMBINE module. *Integration of technical or biological replicates.*

The Merging of experiments is done within NORCOMBINE module using the “cardenio” program from the SanXoT software package [1]. “Cardenio” is used to generate suitable relation tables to make averages from technical or biological replicates. Then, NORCOMBINE normalizes ("center") the data integrating the grouped-level to take into account the systematic quantitative error of each experiment.

More in detail

This module needs a set of files, produced by INTEGRATE module:

- “lowerNormV”: this file contains three columns:
  - the identifiers of the lower level,
  - the second column contains the Xinf - Xsup (i.e. the ratios of the lower level, but centred for each element they belong to), and
  - the third column is the former untouched V(inferior) weight.

- “lowerNormW”: this file contains three columns. The first two are the same than “lowerNormV” but the third is the new weight W(inferior) containing the variance of the integration.

More detailed information may be found in “sanxot” integration from SanXoT software package [1].

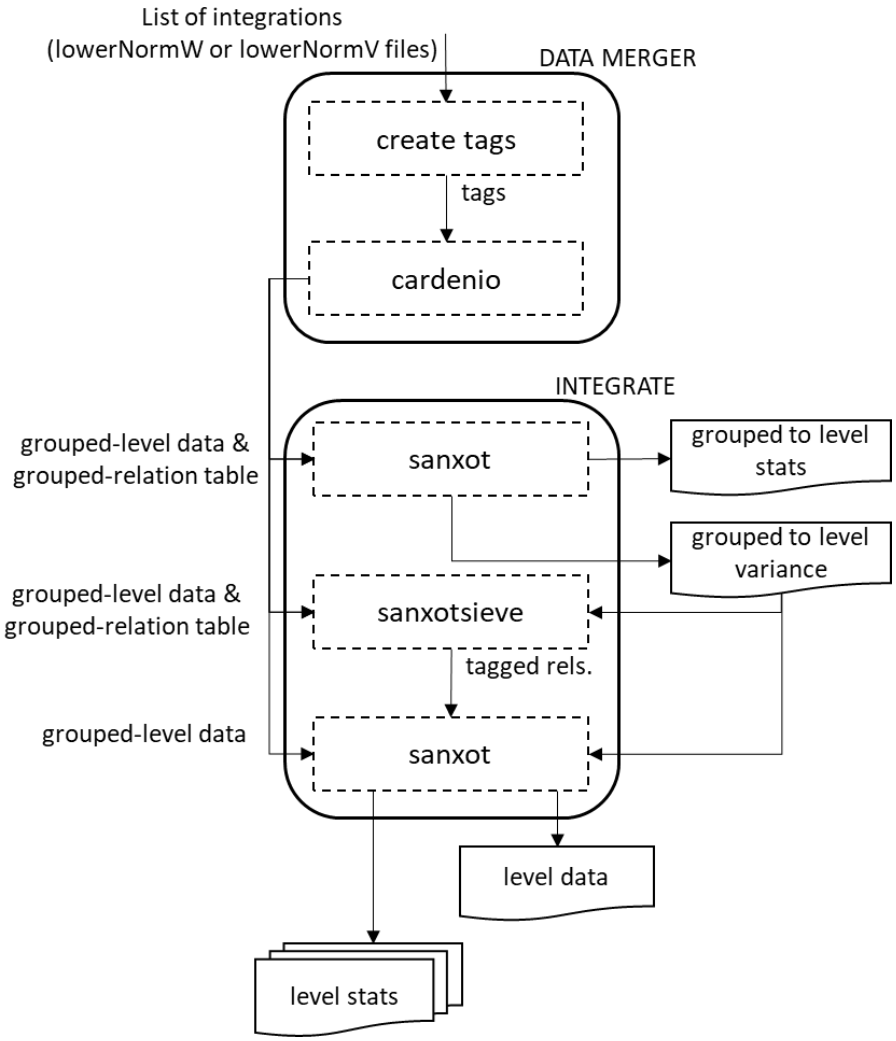


Figure 10. Flowchart of NORCOMBINE module.

Simple parameters

The parameters/columns of this module are:

- Forced execution: checkbox that determines to force the execution or not.
- Sample folder(s): indicates the names of the folders where the lower-level data is located.
- Level: indicates the name of level. The module will use the data file with the same name of level.
- Norm: specifies the normalization scheme (“proteinall” here) to be used with level.
- lowerNorm: selects the type of lower normalization to use: V (the untouched V weight) or W (the new weight containing the variance of the integration).
- Output Sample folder: the name of the folder where the resulting be saved.

	F. exec.	Sample folder(s)	Level	Norm	lowerNorm	Output Sample folder
1	☐	Junkat_WT/*	protein	proteinall	lowerNormV	WT
2	☐	Junkat_KO/*	protein	proteinall	lowerNormV	KO
3	☐					

Figure 11. Task-Table example for NORCOMBINE module. *In this case, the asterisk wildcard has been used to select multiple sample folders.*

Advanced parameters

This module accepts advanced parameters belonging to the integration part:

- Tag: label that below to accept or discard elements is the calculation of variance. By default, the "outliers" are discarded.
- FDR: limit of False Discovery Rate different than 0.01 (1%). If FDR = 0, then the "outliers" are not discarded.
- Var(x): force a variance. By default, the variance calculated by first "sanxot" is applied.
- More params: allows to add more parameters to the programs inside the program. For more detail read "More params" in "Special Parameters" section.

	F. exec.	Sample folder(s)	Level	Norm	lowerNorm	Output Sample folder	Tag	FDR	Var(x)	More params
1	☐	Junkat_WT/*	protein	proteinall	lowerNormV	WT				
2	☐	Junkat_KO/*	protein	proteinall	lowerNormV	KO				
3	☐									

Figure 12. Task-Table with advanced parameters for NORCOMBINE module.

RATIOS

This module calculates the statistical weight of the newly calculated log2-ratios from two comparison samples. This module takes (X,V) of given level from the numerator and denominator samples. It calculates the difference for the new X value (i.e. log2 of the indicated ratios), and combines the V values using the indicated method. Then, the result (X,V) is stores as the same level in the output sample folder.

More in detail

This module takes calculates the new (X,V) values from the numerator and denominator samples. The new X value is the difference between both X's values. The "V method" determines how to calculate the new V value. These methods are:

- the maximum of both V's.
- the applied form V = 1/(1/V1+1/V2).
- and the average of both V's.

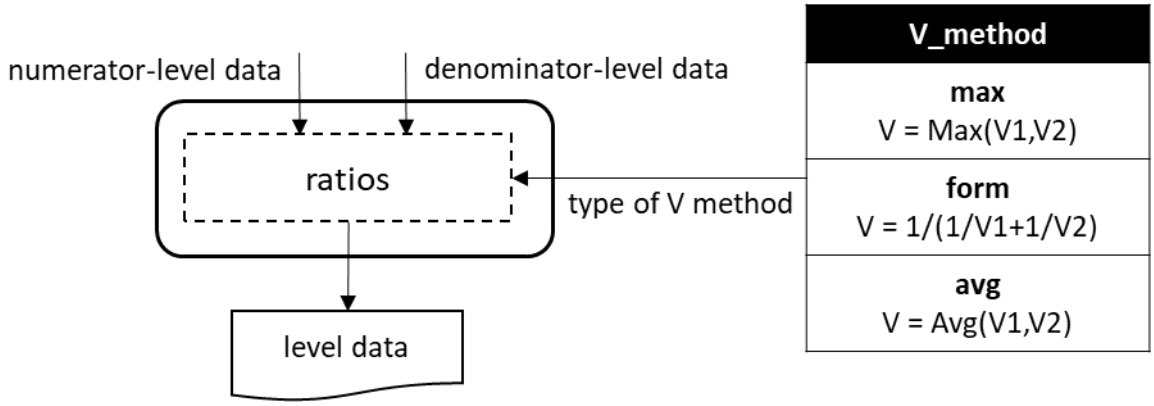


Figure 13. Flowchart of RATIOS module.

Simple parameters

The parameters/columns of this module are:

- Forced execution: checkbox that determines to force the execution or not.
- Ratio numerator: indicates which sample contains the level required to make up the numerator of the ratio to be calculated ("KO" in this case).
- Ratio denominator: indicates which sample contains the level file required to make up the denominator of the ratio to be calculated ("WT" in this case).
- Level: specifies the level from where the ratio calculation should be made.
- Output Sample folder: the name of the folder where the resulting log2-ratio and statistical weight values will be saved ("KO_vs_WT" in our example).

	F. exec.	Ratio numerator column	Ratio denominator column(s)	Level	V Method	Output Sample folder
1	☐	KO	WT	protein	max	KO_vs_WT
2	☐					

Figure 14. Task-Table of RATIOS module.

SBT

This module is based on The Systems Biology Triangle model [2]. The SBT model performs a triangular integration using lower level, intermediate level and higher level. It is common the levels would be protein, category as lower and intermediate level, and the higher level would be the grand mean. From this way, the SBT module performs a Systems biology analysis.

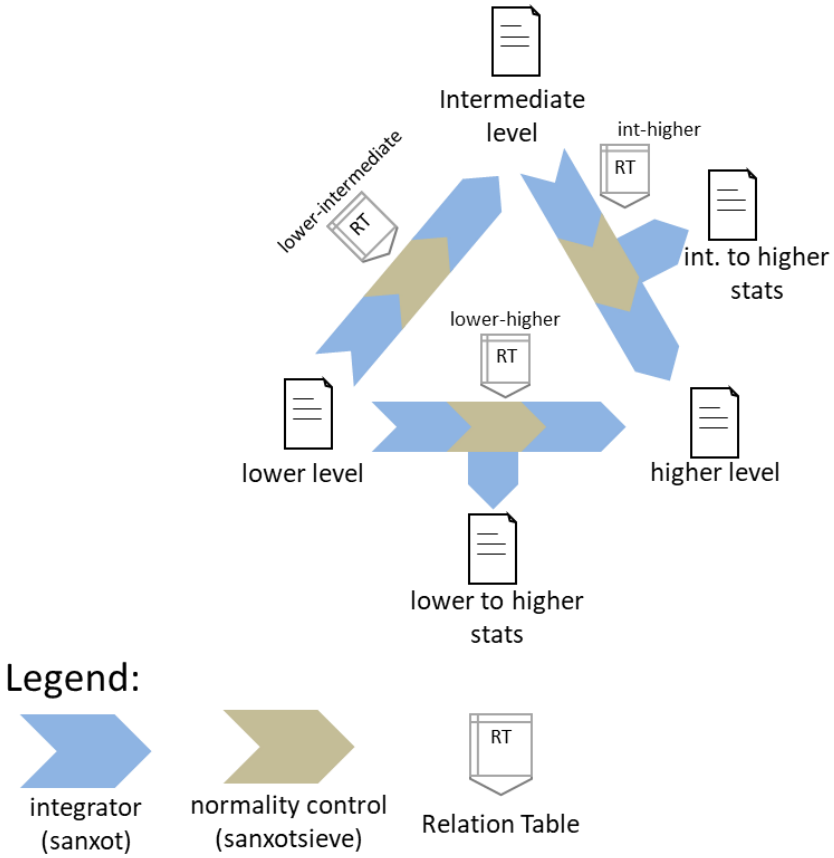


Figure 15. Schema of SBT module.

Simple parameters

The parameters/columns of this module are:

- Forced execution: checkbox that determines to force the execution or not.
- Sample folder(s): indicates the names of the folders where the levels data are located.
- Lower level: indicates the name of lower-level ("protein" in this case).
- Intermediate level: indicates the name of intermediate-level ("category").

By default, the higher-level is the grand mean of each level.

	F. exec.	Sample folder(s)	Lower level	Intermediate level
1	☐	KO_vs_WT	protein	category
2	☐			

Figure 16. Task-Table of SBT module.

Advanced parameters

This module accepts advanced parameters belonging to the integration part:

- Output Sample folder: the name of the folder where the resulting log2-ratio and statistical weight values will be saved ("KO_vs_WT" in our example).
- Lower-Higher level and Int(ermediate)-Higher level: in the case you don't want to use the grand mean for the higher level, you can the level for the respective integration.

- Low(er) > Int(ermediate) Tag and Int(ermediate) > Hig(her) Tag: labels from lower level to intermediate level and from intermediate level to higher level respectively that below to accept or discard elements is the calculation of variance. By default, the “outliers” are discarded.
- Low(er) > Int(ermediate) FDR and Int(ermediate) > Hig(her) FDR: limit of False Discovery Rate different than 0.01 (1%) for the from lower level to intermediate level and from intermediate level to higher level respectively. If FDR = 0, then the “outliers” are not discarded.
- Low(er) > Int(ermediate) Var(x) and Int(ermediate) > Hig(her) Var(x): force a variance for the from lower level to intermediate level and from intermediate level to higher level respectively. By default, the variance calculated by first “sanxot” is applied.
- More params: allows to add more parameters to the programs inside the program. For more detail read “More params” in “Special Parameters” section.

Figure 17. Task-Table with advanced parameters for SBT module.

Report modules

REPORT

The REPORT module allows the collection of the integration data into a result table. It compiles the statistical results from the given integration and stores the values in a single table.

Simple parameters

The parameters/columns of this module are:

- Forced execution: checkbox that determines to force the execution or not.
- Sample folder(s): indicates the names of the folders where the statistical data for each integration is located.
- Lower level: indicates the starting level (peptide, protein, category) for the integration whose data are to be reported.
- Higher level: indicates the ending level for the integration whose data are to be reported.
- Reported vars: specifies which statistical variables will be reported. The available variables are “n”, “tags”, “Z”, “FDR”, “Xsup”, “Vsup”, “Xinf”, “Vinf”.
- Output report: is the report filename (without extension).

	F. exec.	Sample folder(s)	Lower level	Higher level	Reported vars	Output report
1	☐	*	scan	peptide	n	Nscan_pep
2	☐	*	peptide	protein	Xinf,Z,FDR	Nscan_Normpep_prot_XZ
3	☐	*	protein	proteinall	Xinf,Z,FDR	Nscan_Normpep_Quanprot_XZ
4	☐	*	peptide	peptideall	Z,FDR	Nscan_Quanpep
5	☐	*	peptide	protein	n	Nscan_Quanpep_prot
6	☐	*	protein	proteinall	Z,FDR	Nscan_Quanpep_Quanprot_Filt
7	☐	*	peptide	protein	n	Npep_prot
8	☐	*	protein	proteinall	Z,FDR	Npep_Quanprot
9	☐	*	protein	category	n	Nprot_cat
10	☐	*	category	categoryall	Z,FDR	Nprot_Quancat
11	☐	*	protein	category	n	Npep_Quanprot_cat
12	☐	*	category	categoryall	Z,FDR	Npep_Quanprot_QuanCat_Filt
13	☐					

Figure 18. Task-Table of REPORT module.

The first row of above task-table takes the "n" variable from the statistical data of "scan-peptide" integration (scan2peptide_outStats.tsv file). This means, from the indicated lower level and higher level. Due to asterisk wildcard, the report of this row will contain all samples of the experiments. The output of this report will be saved into "Nscan_pep" filename in the "reports" folder of your project.

The second row takes the "Xinf","Z", and "FDR" variables from the statistical data of "peptide-protein" integration. Also, this row will create a report file for all samples of the experiments, called "Nscan_Normpep_prot_XZ".

The description of the rest of rows are similar.

Advanced parameters

This module accepts advanced parameters:

- Level names to show: if you don't want to show one of the level data in the report. By default, all the levels are shown.
- Merge with report: Join with designates the file whose Reported vars will be incorporated into Output after intersection with the latter file.
- Add columns from relation table: insert into the current report the columns from the relation table (RT) at the end. The RT columns are added based on the values of the "lower level". It is possible to use multiple RT separated by comma.
- Filter: the data of report could be filtered based on the reported variables (n, Z, FDR, etc). For more detail read "Filter" in "Special Parameters" section.

Output report	Level nam	Merge with report	Add columns from	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 , protein2descriptio	
Npep_Quanprot_QuanCat_Filt		Npep_Quanprot_cat		FDR_category2categoryall < 0.05 & n_protein2category >= 5 & n_protein2category <= 100

Figure 19. Task-Table with advanced parameters for REPORT module.

The way to merge one report with another is through the column names “Output report” and “Merge with report”. Each row represents one report and the “output report” is the file name of this report. The “merge with report” houses the name of previous report name that will be used to join its own data with the current report.

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 20. Example of join of two reports. *The first row creates a report with the (n)umber of peptides per scan, and this data is saved in “Nscan_pep” file. The second row creates a report file called “Nscan_Normpep_protXZ” with the variables “Xinf”, “Z”, and “FDR” for the integration peptide to protein. In addition, the (n)umber of peptides per scan that are saved in the “Nscan_pep” report file, will join to this report based on the values of “peptide” (lower level).*

In addition, it is possible to add more data into the report from one or more relation tables (RTs). The condition for the adding is that the name of lower level is within the column names of RT. If this is the case, the columns from RT are added based on the values of lower level. If not, the REPORT module checks whether the name of higher level is with in the column names of RT. If this is the case, the columns are added based on the values of higher level. Otherwise, the module does nothing.

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 21. Add data into report from relation tables. *The RT "protein2gene" contains one column with the protein identifiers, named "protein", and another column with the gene of each protein. The RT "protein2description" contains also the protein identifiers with a column name "protein" and another column with the description of each protein. The gene names and the protein descriptions contained in these RTs will be included in the "Npep_Quanprot_cat" report file based on the protein identifiers ("lower level").*

Moreover, the data of report file could be filtered from the "Filter" parameter based on logical conditions over the reported variables. The filtering could be as in these examples:

- n_protein2category == 5, retrieves the data report which the (n)umber of categories per protein is equal to 5.
- n_protein2category <= 100, retrieves the data which the (n)umber of categories per protein is less than or equal to 100.
- n_protein2category >= 5 & n_protein2category <= 100, retrieves the data which the (n)umber of categories per protein is greater or equal than 5 and less or equal than 100.
- KO_vs_WT@FDR_category2categoryall < 0.05, retrieves the data which the FDR of category to categoryall integration from "KO_vs_WT" samples is less than 5%.

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 22. Filtering the report data.

The variables in the filtering have a standard composition. In the following example, "n_protein2category", the variable is composed with the reported variable (n) and the integration (protein2category). This filter is applied for the protein to category integration of all samples.

However, in the example of "KO_vs_WT@FDR_category2categoryall", the filter is applied in the "FDR" variable of category to categoryall integration but only for the "KO_vs_WT" sample folder. Moreover, in the example "WT1,WT2@FDR_category2categoryall" the filter is applied for the "WT1" and "WT2" samples.

SANSON

The SANSON module detects the categories that enclose a similar set of proteins, and then, it shows the changing proteins within each category generating a similarity graph.

Simple parameters

The parameters/columns of this module are:

- Forced execution: checkbox that determines to force the execution or not.
- Sample folder(s): indicates the names of the folders where the lower-level data is located.
- Lower level: indicates the name of lower-level ("protein" in this case).
- Higher level: indicates the name of higher-level to integrate ("category" in this case).
- Output Sample folder: the name of the folder where the resulting be saved.

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

Figure 23. Task-Table of SANSON module.

Advanced parameters

This module accepts advanced parameters:

- Lower norm: specifies the normalization scheme to be used with the lower level.
- Higher norm: specifies the normalization scheme to be used with the higher level.
- Tag: filter the label of tags. By default, the "out" tag coming from the outliers is filtered.
- Filter: the data of report could be filtered based on the reported variables (n, Z, FDR, etc). For more detail read "Filter" in "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 24. Task-Table with advanced parameters for SANSON module.

Complete Modules

WSPP-SBT

WSPPG-SBT

WPP-SBT

Special parameters

Keywords for the task-tables

The task-tables accept some keywords that represent values mostly related to the working project. The keywords are the following:

- __IDQFIL__ represents the Identification-Quantification file (ID-q file). This keyword is used mainly by the task-table of RELS CREATOR module. When the workflow is using an adaptor and we don't know "a priori" the ID-q created by the adaptor, we use this __IDQFIL__ keyword in the "Table from which RT is extracted" parameter.

Multiple samples

It is possible to add multiple samples within the "Sample folder(s)" parameter in the task-tables. For example, having the following task-table for LEVEL CREATOR module:

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 add multiple samples separated by comma.

Sample folder(s)	Lower level	Higher level
Junkat_WT/WT_1 , Junkat_WT/WT_2 , Junkat_WT/WT_3 , Junkat_WT/WT_4 , Junkat_KO/KO_1 , Junkat_KO/KO_2 , Junkat_KO/KO_3 , Junkat_KO/KO_4	peptide	protein
Junkat_WT/WT_1 , Junkat_WT/WT_2 , Junkat_WT/WT_3 , Junkat_WT/WT_4 , Junkat_KO/KO_1 , Junkat_KO/KO_2 , Junkat_KO/KO_3 , Junkat_KO/KO_4	protein	category
Junkat_WT/WT_1 , Junkat_WT/WT_2 , Junkat_WT/WT_3 , Junkat_WT/WT_4 , Junkat_KO/KO_1 , Junkat_KO/KO_2 , Junkat_KO/KO_3 , Junkat_KO/KO_4	peptide	peptideall
Junkat_WT/WT_1 , Junkat_WT/WT_2 , Junkat_WT/WT_3 , Junkat_WT/WT_4 , Junkat_KO/KO_1 , Junkat_KO/KO_2 , Junkat_KO/KO_3 , Junkat_KO/KO_4	protein	proteinall
Junkat_WT/WT_1 , Junkat_WT/WT_2 , Junkat_WT/WT_3 , Junkat_WT/WT_4 , Junkat_KO/KO_1 , Junkat_KO/KO_2 , Junkat_KO/KO_3 , Junkat_KO/KO_4	category	categoryall

Asterisk is our jack of all trades

Apart of include multiple samples separated by comma, there is the asterisk is used as a wildcard. This asterisk means “zero or more characters” for the “Sample folder(s)” parameter in the task-tables. If we have the following task-table for LEVEL CREATOR module:

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 is the ratio saved in its corresponding “Output Sample folder”. E.g. the ratio of 113 with the mean of 113,114,115,116 is saved in “Junkat_WT/WT_1”, the ratio of 114 is saved in “Junkat_WT/WT_2”, and so on. One way to create the task-table of INTEGRATE module could be as illustrate the below table where each row represents a sample and its integration.

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, it is an easy way to do applying the asterisk wildcard. The task-table below allows to select the multiple sample folders. E.g. “Junkat_WT/*” selects all the sample folders within the “Junkat_WT”, and “Junkat_KO/*” selects all the sample folders within the “Junkat_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 only asterisk. The first row of the following task-table takes the “peptide” level for all samples folders of this project, the second row takes the “protein” levels for all sample folders, and so on.

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

More params

Filter param (in REPORT module)

Workflows

Adaptors

Main Input

The Main-Input adaptor provides the following parameters to your project:

- Project folder: describes the path to the folder where iSanXoT output files will be stored.
- Identification file: specifies the location of the file that contains the identification and quantification data.

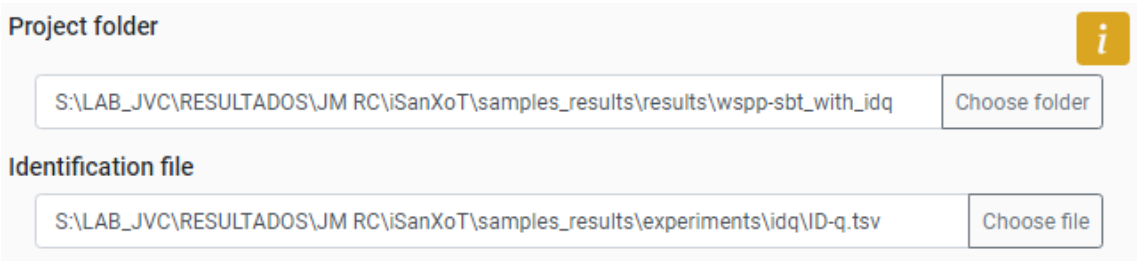


Figure 25. Panel of Main-Input adaptor with a worked-out example.

Close Search – Inputs from PD

This adaptor allows you to transform the output of Proteome Discoverer (PD) into an Identification-Quantification file appropriate for iSanXoT workflows. The need outputs of PD are the tested in PD 2.5,

Table 2. **Data accessed by iSanXoT in the *PSMs.txt* files.**

Header	Description
Spectrum File	Name of the raw LC-MS/MS file
First Scan	Scan number identifier
Sequence	Peptide amino acid sequence
Modifications	Unimod[1] chemical or posttranslational modifications to peptide sequence
Charge	Peptide charge
Search Engine Rank	Rank of search engine
XCorr	Cross-correlation value as provided by SEQUEST[2] algorithm
MHplus in Da	Measured monoisotopic protonated peptide mass in Da
Theo MHplus in Da	Theoretical monoisotopic protonated peptide mass in Da
Delta M in ppm	Difference between measured and theoretical monoisotopic mass in ppm
Protein Accessions	Accession codes for the proteins to which the peptide sequence is ascribed
Protein Descriptions	List of protein descriptions separated by comma
113-121, 126-131, etc	Intensity of reporter ions

Once the program has been successfully installed, executing the isanxot.bat script will bring you to the iSanXoT main page, where a number of predefined workflows are displayed. The link WSPP-SBT sample with iSanXoT databases found below the short description of the Basic Workflow will take you to the workflow Inputs window (Fig. 1), where the following information is provided:

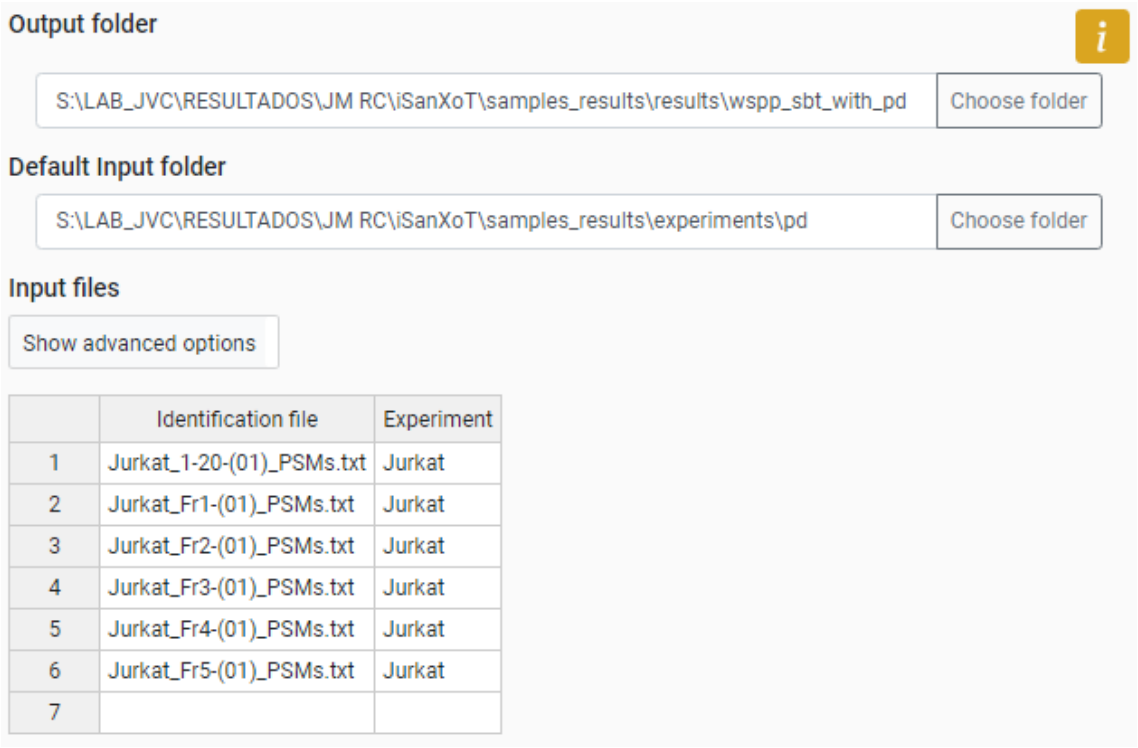
- Input folder specifies the location of the files containing PSM identification and quantification data (the PSMs.txt files in this example);

- Output folder describes the path to the folder where iSanXoT output files will be stored. Selection of an output folder other than the input folder is strongly recommended;

- Select input files indicates which of the PSM.txt files stored in the input folder must be considered by iSanXoT (all six files in this case). PSMs.txt file names are listed under Infile, while their experiment allocation is indicated in Experiment. All six PSMs.txt files originate from the same experiment (termed “TMT”) in our example, but larger projects may encompass several experiments (e.g. “TMT1”, “TMT2” and “TMT3”; or “Exp1”, “Exp2” and “Exp3”). Every time an Input Folder is selected (by clicking on Choose folder), Infile cells are automatically filled in with the names of every single file or subfolder therein; be aware that any file other than those containing PSM identification and quantitation data (e.g. PSMs.txt files) should be removed from Infile.

[1] https://www.unimod.org/

[2] Eng J.K. *et al.* (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom*. **5**, 976–89.

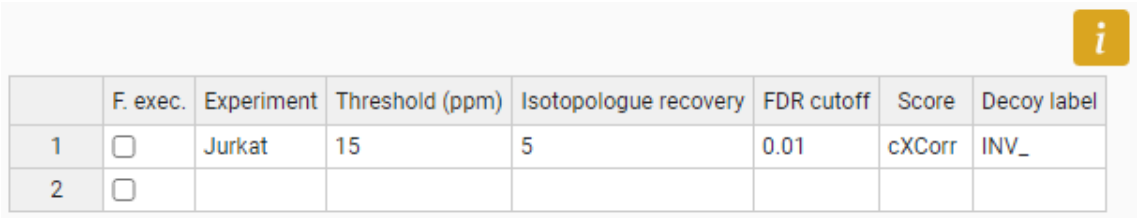


Validating peptide identification with FDR

iSanXoT relies on the probability ratio (pRatio) method, an algorithm that calculates the probability of random peptide matching and provides the corresponding FDR for peptide identification. The FDR parameters displays the following fields:

- Forced execution: checkbox that determines to force the execution or not.
- Experiment: the aforementioned experiment allocation.
- Threshold (ppm): is the postscoring mass filtering cut-off to be applied after using wide mass windows in the database search, as was the case with these sample data. Threshold is actually equivalent to the relative deviation experimentally observed for precursor ions in a particular LC-MS/MS run.
- Isotopologue recovery: allows pRatio to recover precursor m/z values matching some 13C isotopologue when using wide mass windows in the LC-MS/MS acquisition that otherwise would remain unnoticed. For that, it must be indicated whether precursor m/z values should be tracked only around their experimental m/z (value = 1) or also ± 1 Th (value = 3) and ± 2 Th (value = 5, the one used here) away.
- FDR cutoff: establishes the FDR cut-off for PSM validation. The value used here (0.01, i.e. 1% FDR) implies that one in every 100 validated PSMs is incorrect.
- Score: determines the score used to calculate the FDR. These scores are “XCorr” (SEQUEST5 cross correlation score), or “cXCorr” (the corrected XCorr) will be used by pRatio for FDR calculation.

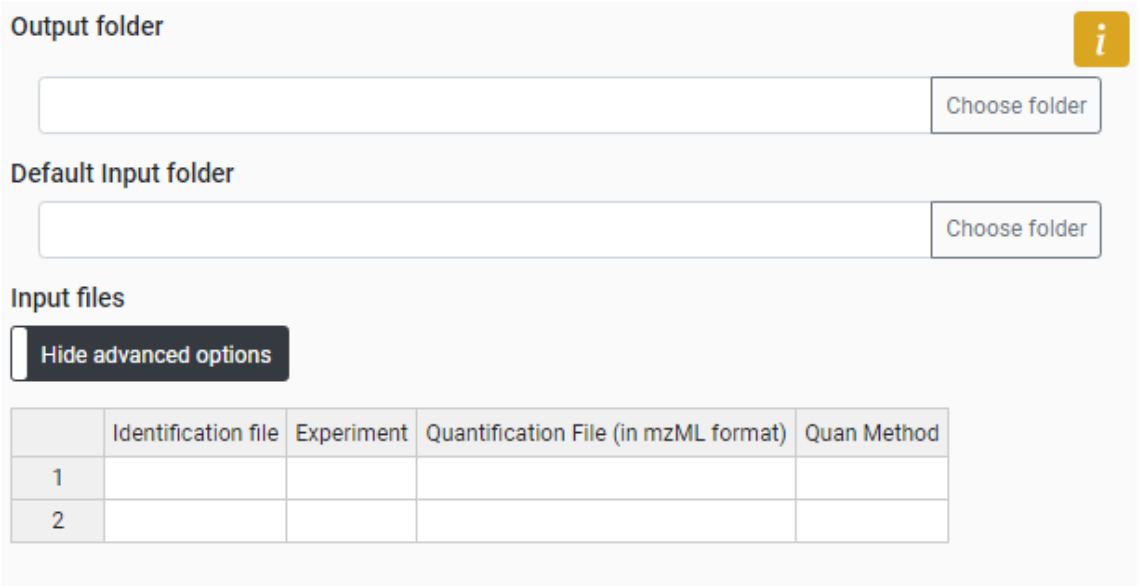
- Decoy label: is the tag attached to decoy protein identifiers in the concatenated protein database previously used for peptide identification ("INV_" in this example).



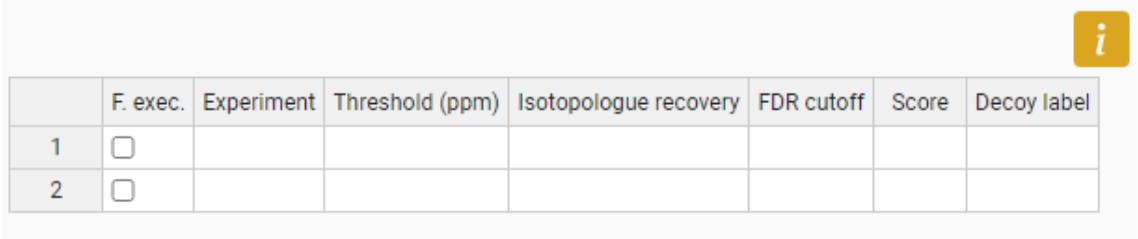
	F. exec.	Experiment	Threshold (ppm)	Isotopologue recovery	FDR cutoff	Score	Decoy label
1	☐	Jurkat	15	5	0.01	cXCorr	INV_
2	☐						

Figure 26. Task-Table of FDR module.

Close Search – Inputs from MSFragger

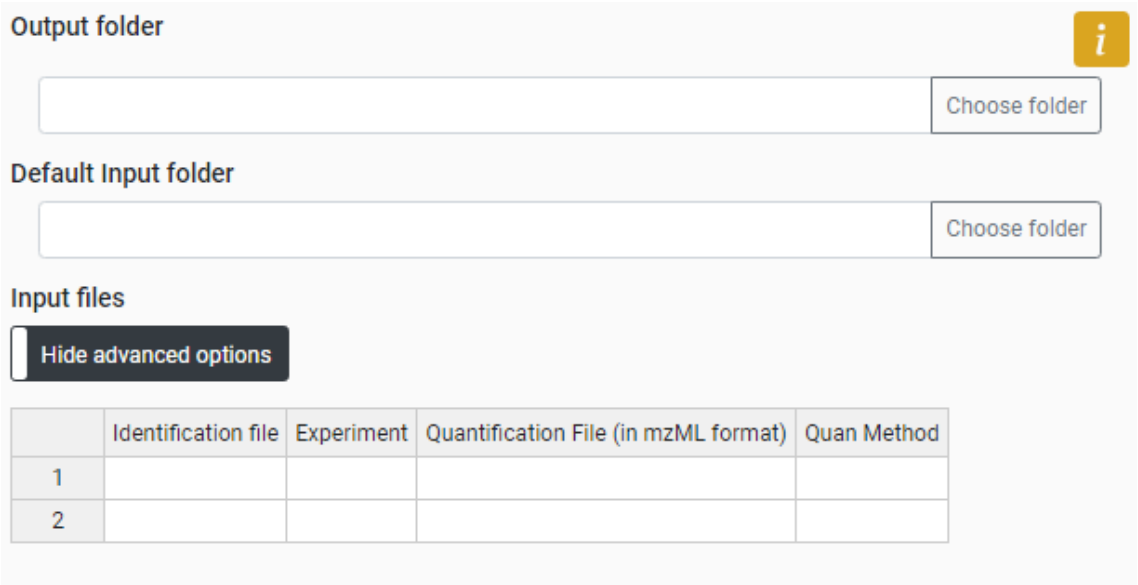


	Identification file	Experiment	Quantification File (in mzML format)	Quan Method
1				
2				

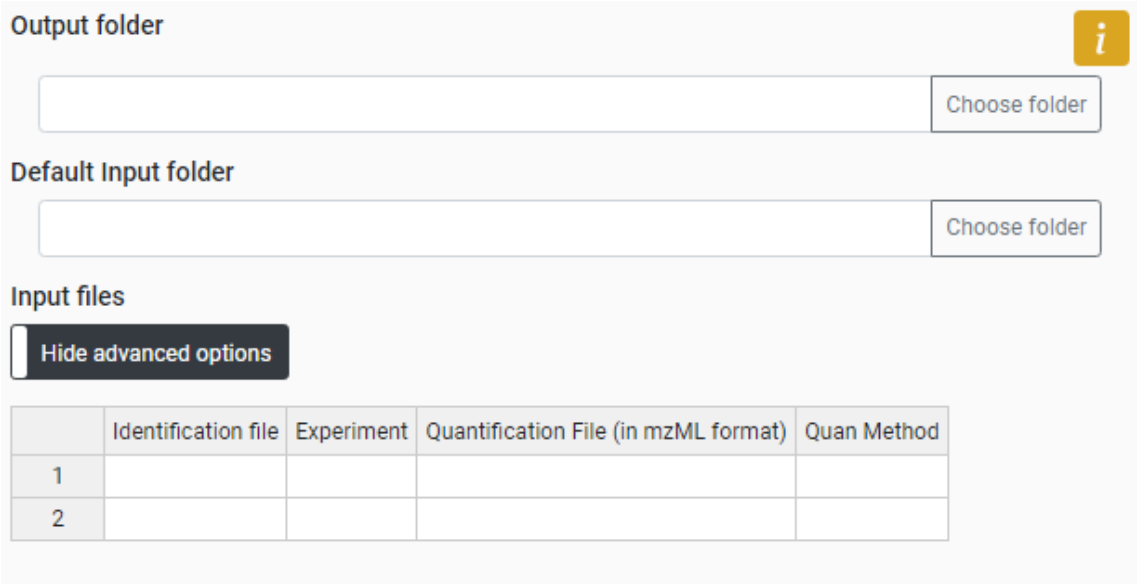


	F. exec.	Experiment	Threshold (ppm)	Isotopologue recovery	FDR cutoff	Score	Decoy label
1	☐						
2	☐						

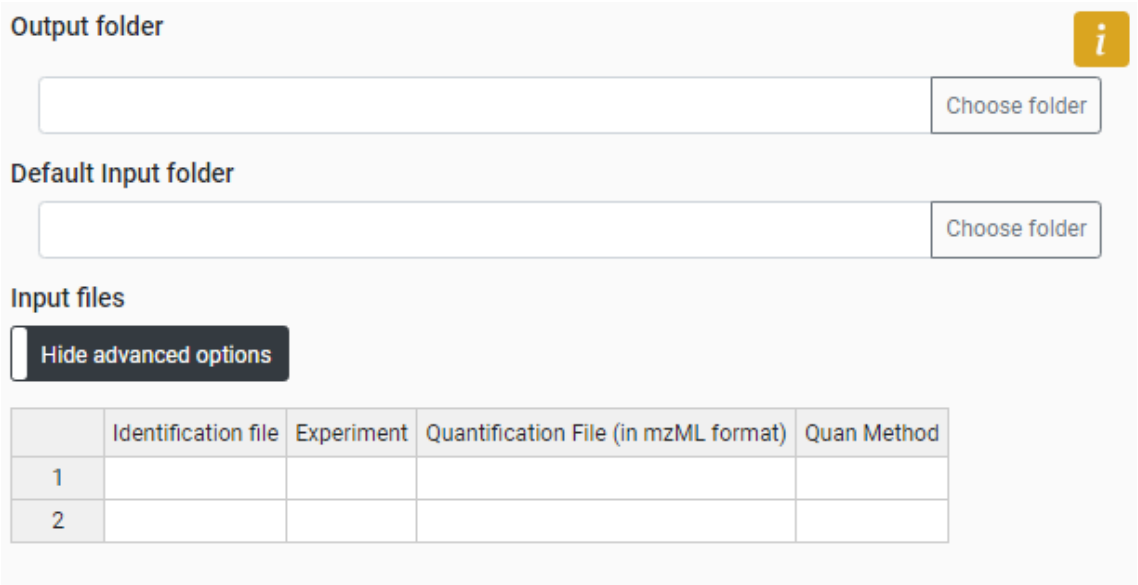
Close Search – Inputs from Comet



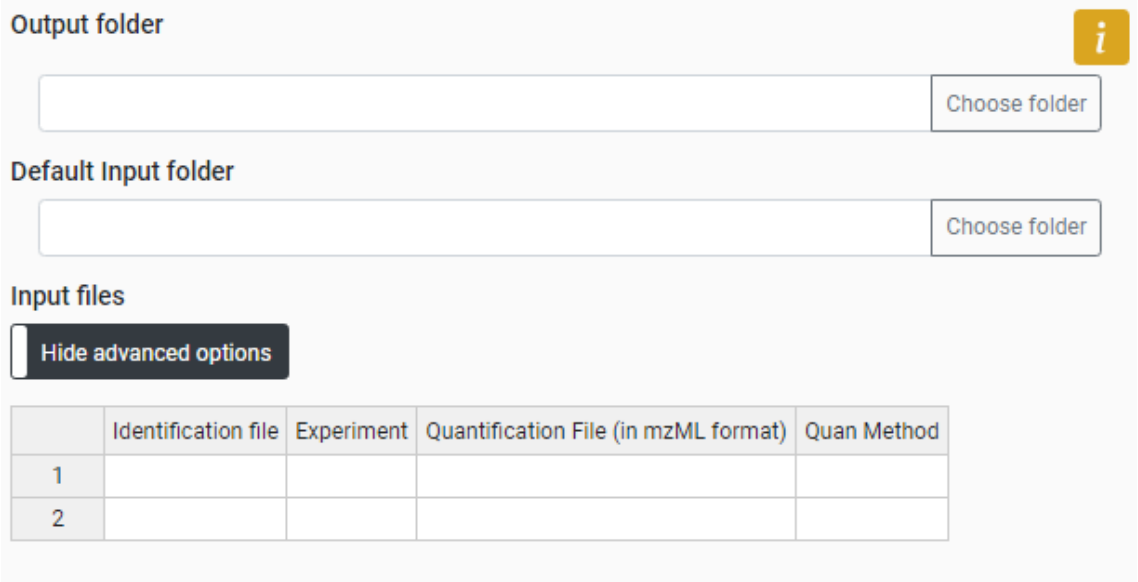
Close Search – Inputs from MaxQuant



Open Search – Inputs from MSFragger



Open Search – Inputs from Comet-PTM



References

[1] Trevisan-Herraz M. et al. (2019) SanXoT: a modular and versatile package for the quantitative analysis of high-throughput proteomics experiments. Bioinformatics. 35, 1594-96.

[2] Garcia-Marques, F., et al., A Novel Systems-Biology Algorithm for the Analysis of Coordinated Protein Responses Using Quantitative Proteomics. Mol Cell Proteomics, 2016. 15(5): p. 1740-60.

[3] Navarro, P., et al., General statistical framework for quantitative proteomics by stable isotope labeling. J Proteome Res, 2014. 13(3): p. 1234-47.