Method S3. Manual of the Standalone Version of ANPELA 2.0.

TITLE: ANPELA 2.0: Significantly Enhanced Quantification Tool for Cytometry-based Single-cell Proteomics

DESCRIPTION: ANPELA is an online tool that specializes in the analysis and performance assessment of proteome quantification. The latest standalone ANPELA 2.0 enables the quantification of cytometry-based single-cell proteomics (CySCP) data and the performance assessment of each preprocessing workflow from multiple perspectives for cell subpopulation identification (CSI) and pseudo-time trajectory inference (PTI) studies. Particularly, it **(*a*)** describes the first systematic workflow for quantifying raw SCP data generated by both flow and mass cytometry (FC/MC), **(*b*)** assesses all possible workflows (~1125 random combinations of compensation, transformation, normalization and signal clean methods) based on their performances, **(*c*)** provides the unique function of identifying the proper workflow(s) for the studied dataset by comprehensively ranking over 1,000 available workflows, **(*d*)** visualizes the resulting quantification performance of each workflow and **(*e*)** exports the quantified CySCP data in the format of FCS and/or expression count matrix. All in all, this tool makes the performance assessment of whole CySCP data workflow possible (collectively assessed by four well-established criteria with distinct underlying theories) and gives the ranking results of all possible workflows based on the criteria preferred and selected by the users.

OPERATION PROCEDURE

1. Download and Install the Latest Release of *R* and *RStudio*

The latest version of ANPELA 2.0 was developed and tested on '3.6.1' version of ***R***. The suitable version of ***R*** for Windows and its corresponding integrated development environment ***RStudio*** can be downloaded directly from (***1***) the ANPELA 2.0 website:

http://idrblab.cn/anpela2023/R&Rstudio.zip

OR from (***2***) the official website of "*The R Project for Statistical Computing*" (https://cran.r-project.org/) and the official website of RStudio (https://www.rstudio.com/).

The whole installation should be completed by two sequential steps. First, please install the '3.6.1' version of ***R*** by double clicking the executable file (**1-R-3.6.1-win.exe**) and following step-by-step instructions during the whole setup process (find more information on https://www.r-project.org/about.html). Second, install the ***RStudio*** by double clicking the executable file (**2-RStudio-2022.02.1-461.exe**) and following the step-by-step instructions during the whole setup process.

2. Download the Source Code of the Standalone Version of ANPELA 2.0

The source code of ANPELA 2.0 together with the supporting ***R*** packages can be downloaded from:

http://idrblab.cn/anpela2023/ANPELA2.0Sourcecode.zip

Please decompress it by right clicking and selecting the "Extract to anpela\" in **Your Preferred Directory**.

3. Set the Runtime Environment

3.1 Set Working Directory

After running ***RStudio***, please change your working directory (in ***RStudio*** environment) to "**Your Preferred Directory**\anpela\ANPELA 2.0_Sourcecode\" by typing and then running the following ***R*** command:

```
setwd("Your Preferred Directory/anpela/ANPELA 2.0_Sourcecode")
```

NOTE: (1) the ***R*** environment uses **forward slash (/)** to indicate the filepath, which is different from the Windows CMD commands (backslash); (2) user can confirm your current working directory again by typing and then running the following ***R*** command:

```
getwd()
```

3.2 Set Library Paths

All of the ***R*** packages that ANPELA 2.0 depends on are provided in the folder of "Your Preferred Directory/anpela/ANPELA 2.0_Sourcecode/library". Please set the library trees within which packages are looked for by typing and then running the following ***R*** command:

```
.libPaths(c("./library", .libPaths()))
```

3.3 Load Required Scripts

All of the ***R*** scripts which load required packages and define a number of functions are provided in the folder of "Your Preferred Directory/anpela/ANPELA 2.0_Sourcecode/src". Please load these scripts by typing and then running the following ***R*** command:

```
sapply(paste0("./src/", list.files(path = "./src", recursive = T, pattern = ".R$")), source)
```

4. Conduct Systematic Workflows for Quantifying Raw SCP Data Generated by FC/MC

Please quantify your raw SCP data acquired from FC and MC by running the function of "***FCquan***" and "***MCquan***" respectively. Particularly, as the output of these two functions is exactly the input of other functions in ANPELA 2.0 except "***ranking***", it is recommended to save the output of these two functions as .RData file, restart the R session and then load it when needed in order to avoid memory explosion. The detailed information about the argument of these two functions is provided in **Table S4.** Moreover, as the compensation method of "AutoSpill", "CATALYST" and "FlowCore" required additional files to assist quantification, the sample files are given in the folder of "Your Preferred Directory/anpela/ANPELA 2.0_Sourcecode/exampler".

5. Analysis and Performance Assessment of the Workflow(s)

Please assess all workflows which are used while running the function of "***FCquan***" or "***MCquan***" based on their performances in CSI and PTI studies by running the function of "***Classassess***" and "***TIassess***" respectively. Considering that the subsequent overall ranking is

based on collective considerations of representative assessing metric values and their corresponding performance grades distinguished by well-defined cutoffs under all criteria, the representative assessing metric under each criterion is recommended in this step. Particularly, as the output of these two functions is exactly the input of the function of "***ranking***", it is recommended to save the output of these two functions as .RData file, restart the R session and then load it when needed in order to avoid memory explosion. The detailed information about the argument of these two functions is provided in **Table S4**.

6. Identify the Proper Workflow(s) for the Studied Dataset by Comprehensive Ranking

Please rank all workflows assessed by the function of "***Classassess***" or "***TIassess***" by running the function of "***ranking***". The detailed information about the argument and output of this function is provided in **Table S4** and **Table S5** respectively.

7. Visualize the Resulting Quantification Performance of Each Workflow

Please visualize the resulting quantification performance of each workflow for CSI and PTI studies by running the function of "***Classplot***" and "***TIplot***" respectively. The detailed information about the argument and output of these two functions is provided in **Table S4** and **Table S5** respectively.

8. Export the Quantified CySCP Data Generated by the Specified Workflow(s)

Please export the quantified CySCP data in the format of FCS and/or expression count matrix (markers in column and events/cells in row) by running the function of "***exportFCS***". The detailed information about the argument and output of this function is provided in **Table S4** and **Table S5** respectively.

Table S4. A comprehensive list of functions provided in ANPELA 2.0 together with their descriptions. For each function, its argument name, value type, argument description, allowable argument values and the default value are listed.

(func1). FCquan()

***Description*:** this function enables the quantification of raw SCP data acquired from FC by at most ~720 available workflows (each workflow is distinct by randomly combining methods of compensation, transformation, normalization and signal clean), which facilitates the subsequent application of performance assessment, ranking and plotting.

Argument	Type	Description of the Argument and the Allowable Values	Default
datapath	vector *(character)*	The absolute path of the folder storing the FCS raw data files.	/
metadata	vector *(character)*	The absolute filepath of the metadata file. *The exampler for the study type of "CSI" and "PTI" are provided with the name of "metadata(CSI).csv" and "metadata(PTI).csv" in the folder of "exampler" respectively.*	/
studytype	vector *(character)*	The type of your study, including "CSI" (Cell Subpopulation Identification) and "PTI" (Pseudo-time Trajectory Inference).	/
mergeM	vector *(character)*	The method of merging multiple FCS files. When multiple FCS files are selected, cells can be combined using one of the four different methods including "Ceil", "All", "Min" and "Fixed". *Ceil: up to a fixed number (specified by fixedNum) of cells are sampled without replacement from each FCS file and combined for analysis.* *All: all cells from each FCS file are combined for analysis.* *Min: The minimum number of cells among all the selected FCS files are sampled from each FCS file and combined for analysis.* *Fixed: a fixed num (specified by fixedNum) of cells are sampled (with replacement when the total number of cell is less than fixedNum) from each FCS file and combined for analysis.*	"Fixed"
fixedNum	vector *(numeric)*	The fixed number of cells to be extracted from each FCS file.	200

compensationM	vector *(character)*	The method(s) of compensation for flow cytometry data, including "AutoSpill", "FlowCore", "MetaCyto" and "None".	a vector of all methods
transformationM	vector *(character)*	The method(s) of transformation for flow cytometry data, including "Arcsinh Transformation", "Asinh with Non-negative Value", "Asinh with Randomized Negative Value ", "Biexponential Transformation", "Box-Cox Transformation", "FlowVS Transformation", "Hyperlog Transformation", "Linear Transformation", "LnTransform", "Log Transformation", "Logicle Transformation", "QuadraticTransform", "ScaleTransform", "TruncateTransform" and "None".	a vector of all methods
normalizationM	vector *(character)*	The method(s) of normalization for flow cytometry data, including "GaussNorm", "WarpSet" and "None".	a vector of all methods
signalcleanM	vector *(character)*	The method(s) of signal clean for flow cytometry data, including "FlowAI", "FlowClean", "FlowCut" and "None".	a vector of all methods
spillpath	vector *(character)*	The filepath(s) to the .fcs file(s) of compensation beads or cells. The spillover information for a particular experiment is often obtained by running several tubes of beads or cells stained with a single color that can then be used to determine a spillover matrix for use. *NOTE: Only needed when "FlowCore" is included in the argument of "compensationM". The filenames of these .fcs files must correspond to the names of stain channels. If the original .fcs file contains a pre-calculated spillover matrix as the value of the $SPILLOVER, $spillover or $SPILL keywords, this can be set as NULL.*	NULL
FSC	vector *(character)*	The name of the forward scatter parameter. *NOTE: Only needed when "FlowCore" is included in the argument of "compensationM".*	"FSC-H"
SSC	vector *(character)*	The name of the side scatter parameter. *NOTE: Only needed when "FlowCore" is included in the argument of "compensationM".*	"SSC-H"
control.dir	vector *(character)*	The absolute path of the folder storing FCS files of single-color controls. *NOTE: Only needed when "AutoSpill" is included in the argument of "compensationM".*	NULL
control.def.file	vector *(character)*	The absolute filepath of your .csv file defining the filenames and corresponding channels of the single-color controls. *NOTE: Only needed when "AutoSpill" is included in the argument of "compensationM".*	NULL
logbase	vector *(numeric)*	The base of the Log Transformation. *NOTE: Only needed when "Log Transformation" is included in the argument of*	10

		"transformationM".	
b1	vector *(numeric)*	The cofactor of Arcsinh Transformation. *NOTE: Only needed when "Arcsinh Transformation" is included in the argument of "transformationM".*	1/150
b2	vector *(numeric)*	The cofactor of Asinh with Non-negative Value. *NOTE: Only needed when "Asinh with Non-negative Value" is included in the argument of "transformationM".*	1/150
b3	vector *(numeric)*	The cofactor of Asinh with Randomized Negative Value. *NOTE: Only needed when "Asinh with Randomized Negative Value" is included in the argument of "transformationM".*	1/150
Quadratica	vector *(numeric)*	The quadratic coefficient "a" in equation y = a*x^2 + b*x + c. *NOTE: Only needed when "QuadraticTransform" is included in the argument of "transformationM".*	1
Quadraticb	vector *(numeric)*	The linear coefficient "b" in equation y = a*x^2 + b*x + c. *NOTE: Only needed when "QuadraticTransform" is included in the argument of "transformationM".*	1
Quadraticc	vector *(numeric)*	The intercept "c" in equation y = a*x^2 + b*x + c. *NOTE: Only needed when "QuadraticTransform" is included in the argument of "transformationM".*	0
lineara	vector *(numeric)*	The multiplicative factor "a" in equation y = a*x+b. *NOTE: Only needed when "Linear Transformation" is included in the argument of "transformationM".*	2
linearb	vector *(numeric)*	The additive factor "b" in equation y = a*x+b. *NOTE: Only needed when "Linear Transformation" is included in the argument of "transformationM".*	0
Truncatea	vector *(numeric)*	The value at which to truncate. *NOTE: Only needed when "TruncateTransform" is included in the argument of*	1

		"transformationM".	
DEM	vector *(character)*	The marker indexes for data processing and performance evaluation. *It is a string separated by comma. For example, "CD43(FITC.A), CD34(APC.A), CD90(BV421.A), CD45RA(BV510.A)". Or you can wait for a prompt while running the program and then input this argument.*	NULL
cores	vector *(numeric)*	The number of CPU cores to be employed for performing parallel computing. *To avoid memory explosion due to parallel computing, the default is the largest integers not greater than half of the number of CPU cores on the current host.*	floor(detectCores()/2)

(func2). MCquan()

***Description*:** this function enables the quantification of raw SCP data acquired from MC by at most ~540 available workflows (each workflow is distinct by randomly combining methods of compensation, transformation, normalization and signal clean), which facilitates the subsequent application of performance assessment, ranking and plotting.

Argument	Type	Description of the Argument and the Allowable Values	Default
datapath	vector *(character)*	The absolute path of the folder storing the FCS raw data files.	/
metadata	vector *(character)*	The absolute filepath of the metadata file. *The exampler for the study type of "CSI" and "PTI" are provided with the name of "metadata(CSI).csv" and "metadata(PTI).csv" in the folder of "exampler" respectively.*	/
studytype	vector *(character)*	The type of your study, including "CSI" (Cell Subpopulation Identification) and "PTI" (Pseudo-time Trajectory Inference).	/
mergeM	vector *(character)*	The method of merging multiple FCS files. When multiple FCS files are selected, cells can be combined using one of the four different methods including "Ceil", "All", "Min" and "Fixed". *Ceil: up to a fixed number (specified by fixedNum) of cells are sampled without replacement from each FCS file and combined for analysis.* *All: all cells from each FCS file are combined for analysis.*	"Fixed"

		Min: The minimum number of cells among all the selected FCS files are sampled from each FCS file and combined for analysis. *Fixed: a fixed num (specified by fixedNum) of cells are sampled (with replacement when the total number of cell is less than fixedNum) from each FCS file and combined for analysis.*	
fixedNum	vector *(numeric)*	The fixed number of cells to be extracted from each FCS file.	200
compensationM	vector *(character)*	The method(s) of compensation for mass cytometry data, including "CATALYST", "CytoSpill" and "None".	a vector of all methods
transformationM	vector *(character)*	The method(s) of transformation for mass cytometry data, including "Arcsinh Transformation", "Asinh with Non-negative Value", "Asinh with Randomized Negative Value", "Biexponential Transformation", "Box-Cox Transformation", "FlowVS Transformation", "Hyperlog Transformation", "Linear Transformation", "LnTransform", "Log Transformation", "Logicle Transformation", "QuadraticTransform", "ScaleTransform", "TruncateTransform" and "None".	a vector of all methods
normalizationM	vector *(character)*	The method(s) of normalization for mass cytometry data, including "Bead-based Normalization", "GaussNorm", "WarpSet" and "None".	a vector of all methods
signalcleanM	vector *(character)*	The method(s) of signal clean for mass cytometry data, including "FlowAI", "FlowCut" and "None".	a vector of all methods
single_pos_fcs	vector *(character)*	The absolute filepath of the .fcs file containing stained samples and control antibody-capture beads/pooled single-stained beads. *NOTE: Only needed when "CATALYST" is included in the argument of "compensationM".*	NULL
single_pos_mass	vector *(numeric)*	The numeric masses corresponding to barcode channels. *NOTE: Only needed when "CATALYST" is included in the argument of "compensationM".*	NULL
CATALYSTM	vector *(character)*	The method for solving linear system, "flow" and "nnls". *NOTE: Only needed when "CATALYST" is included in the argument of "compensationM".*	"nnls"

logbase	vector *(numeric)*	The base of the Log Transformation. *NOTE: Only needed when "Log Transformation" is included in the argument of "transformationM".*	10
b1	vector *(numeric)*	The cofactor of Arcsinh Transformation. *NOTE: Only needed when "Arcsinh Transformation" is included in the argument of "transformationM".*	1/5
b2	vector *(numeric)*	The cofactor of Asinh with Non-negative Value. *NOTE: Only needed when "Asinh with Non-negative Value" is included in the argument of "transformationM".*	1/5
b3	vector *(numeric)*	The cofactor of Asinh with Randomized Negative Value. *NOTE: Only needed when "Asinh with Randomized Negative Value" is included in the argument of "transformationM".*	1/5
Quadratica	vector *(numeric)*	The quadratic coefficient "a" in equation y = a*x^2 + b*x + c. *NOTE: Only needed when "QuadraticTransform" is included in the argument of "transformationM".*	1
Quadraticb	vector *(numeric)*	The linear coefficient "b" in equation y = a*x^2 + b*x + c. *NOTE: Only needed when "QuadraticTransform" is included in the argument of "transformationM".*	1
Quadraticc	vector *(numeric)*	The intercept "c" in equation y = a*x^2 + b*x + c. *NOTE: Only needed when "QuadraticTransform" is included in the argument of "transformationM".*	0
lineara	vector *(numeric)*	The multiplicative factor "a" in equation y = a*x+b. *NOTE: Only needed when "Linear Transformation" is included in the argument of "transformationM".*	2
linearb	vector *(numeric)*	The additive factor "b" in equation y = a*x+b. *NOTE: Only needed when "Linear Transformation" is included in the argument of "transformationM".*	0

Truncatea	vector *(numeric)*	The value at which to truncate. *NOTE: Only needed when "TruncateTransform" is included in the argument of "transformationM".*	1
beads_mass	vector *(numeric)*	The masses of the corresponding calibration beads. *NOTE: Only needed when "Bead-based Normalization" is included in the argument of "normalizationM".*	c(140, 151, 153, 165, 175)
DEM	vector *(character)*	The marker indexes for data processing and performance evaluation. *It is a string separated by comma. For example, "CD103(La139Di), CCR6(Pr141Di), CD19(Nd142Di), C-KIT(Nd143Di), CD11b(Nd144Di)". Or you can wait for a prompt while running the program and then input this argument.*	NULL
cores	vector *(numeric)*	The number of CPU cores to be employed for performing parallel computing. *To avoid memory explosion due to parallel computing, the default is the largest integers not greater than half of the number of CPU cores on the current host.*	floor(detectCores()/2)

(func3). Classassess()

***Description*:** this function assesses quantification performance of all workflows which are used while running the function of "FCquan" or "MCquan" based on comprehensive criteria (each with distinct underlying theories) from the perspective of CSI studies.

Argument	**Type**	**Description of the Argument and the Allowable Values**	**Default**
data	list	The resulting R object of "FCquan" or "MCquan" function for the "CSI" study type. *You can directly use the corresponding object stored in R environment, or save it after running the "FCquan" or "MCquan" function and load it when needed.*	/
clusteringM	vector *(character)*	The method of clustering the processed data prior to performance evaluation, including "FlowSOM" and "PhenoGraph".	"FlowSOM"
ncluster	vector *(numeric)*	The number of clusters for meta clustering in FlowSOM. *NOTE: Only needed when the argument of "clusteringM" is selected as "FlowSOM".*	8

Ca_metric	vector *(character)*	The assessing metric under Criterion Ca for the "CSI" study type, including "AUC" and "F1 score".	"AUC"
Cb_metric	vector *(character)*	The assessing metric under Criterion Cb for the "CSI" study type, including "Silhouette coefficient (SC)", "Xie-Beni index (XB)", "Calinski-Harabasz index (CH)", "Davies-Bouldin index (DB)", "purity" and "Rand index (RI)".	"Silhouette coefficient (SC)"
Cc_metric	vector *(character)*	The assessing metric under Criterion Cc for the "CSI" study type, including "relative weighted consistency (CWrel)" and "consistency score (CS)".	"relative weighted consistency (CWrel)"
ntop	vector *(numeric)*	The number of the most differentially expressed markers that are truncated for calculating the CWrel value. *NOTE: Only needed when the argument of "Cc_metric" is selected as "relative weighted consistency (CWrel)". This value must be less than the number of your selected markers.*	the largest integers not greater than the number of your selected markers
cores	vector *(numeric)*	The number of CPU cores to be employed for performing parallel computing. *To avoid memory explosion due to parallel computing, the default is the largest integers not greater than half of the number of CPU cores on the current host.*	floor(detectCores()/2)

(func4). TIassess()

***Description*:** this function assesses quantification performance of all workflows which are used while running the function of "FCquan" or "MCquan" based on comprehensive criteria (each with distinct underlying theories) from the perspective of PTI studies.

Argument	**Type**	**Description of the Argument and the Allowable Values**	**Default**
data	list	The resulting R object of "FCquan" or "MCquan" function for the "PTI" study type. *You can directly use the corresponding object stored in R environment, or save it after running the "FCquan" or "MCquan" function and load it when needed.*	/
TIM	vector *(character)*	The method of trajectory inference for the processed data prior to performance evaluation, including "scorpius_distSpear", "scorpius_distPear", "scorpius_distManh", "slingshot_tSNE", "prinCurves_tSNE", "slingshot_PCA", "slingshot_diffMaps" and "prinCurves_diffMaps".	"scorpius_distSpear"

Cc_metric	vector *(character)*	The assessing metric under Criterion Cc for the "PTI" study type, including "Spearman correlation" and "Kendall Rank Correlation".	"Spearman correlation"
pathwayhierarchy	vector *(character)*	The absolute filepath of the pathway hierarchy file. *The exampler is provided with the name of "Pathway_Hierarchy.csv" in the folder of "exampler".*	NULL
cores	vector *(numeric)*	The number of CPU cores to be employed for performing parallel computing. *To avoid memory explosion due to parallel computing, the default is the largest integers not greater than half of the number of CPU cores on the current host.*	floor(detectCores()/2)

(func5). ranking()

***Description*:** this function ranks all workflows assessed by the function of “Classassess” and “TIassess” based on collective consideration of values and grades (classified by well-defined cutoffs) under each criterion.

Argument	Type	Description of the Argument and the Allowable Values	Default
data	list	The resulting R object of "Classassess" or "TIassess" function. *You can directly use the corresponding object stored in R environment, or save it after running the "Classassess" or "TIassess" function and load it when needed.*	/

(func6). Classplot()

***Description*:** this function realizes the visualization of the resulting quantification performance of each workflow for CSI studies.

Argument	Type	Description of the Argument and the Allowable Values	Default
data	list	The resulting R object of "FCquan" or "MCquan" function for the "CSI" study type. *You can directly use the corresponding object stored in R environment, or save it after running the "FCquan" or "MCquan" function and load it when needed.*	/
clusteringM	vector	The method of clustering the processed data prior to plotting, including "FlowSOM" and	"FlowSOM"

	(character)	"PhenoGraph".	
rankingC	vector *(character)*	The copy of the resulting ranking file of "Classassess" function, where a number of workflows that do not require plotting have been deleted.	NULL
ncluster	vector *(numeric)*	The number of clusters for meta clustering in FlowSOM. *NOTE: Only needed when the argument of "clusteringM" is selected as "FlowSOM" while calling the "FCquan" or "MCquan" function for obtaing "data".*	8
Cb_metric	vector *(character)*	The assessing metric under Criterion Cb for the "CSI" study type, including "Silhouette coefficient (SC)", "Xie-Beni index (XB)", "Calinski-Harabasz index (CH)" and "Davies-Bouldin index (DB)" for plotting *class_B1_plot (Cluster Distribution Plot)*/"purity" and "Rand index (RI)" for plotting *class_B2_plot (Dimension Reduction Plot Colored by Sample Group and Clustering Information)* and *class_B3_plot (Two-way Clustering Plot of Differential Proteins)*.	"Silhouette coefficient (SC)"
Cc_metric	vector *(character)*	The assessing metric under Criterion Cc for the "CSI" study type, including "relative weighted consistency (CWrel)" and "consistency score (CS)".	"relative weighted consistency (CWrel)"
ntop	vector *(numeric)*	The number of the most differentially expressed markers that are truncated for calculating the CWrel value. *NOTE: Only needed when the argument of "Cc_metric" is selected as "relative weighted consistency (CWrel)" while calling the "FCquan" or "MCquan" function for obtaing "data". This value must be less than the number of your selected markers.*	the largest integers not greater than the number of your selected markers
cores	vector *(numeric)*	The number of CPU cores to be employed for performing parallel computing. *To avoid memory explosion due to parallel computing, the default is the largest integers not greater than half of the number of CPU cores on the current host.*	floor(detectCores()/2)

(func7). TIplot()

***Description*:** this function realizes the visualization of the resulting quantification performance of each workflow for PTI studies.

Argument	**Type**	**Description of the Argument and the Allowable Values**	**Default**

data	list	The resulting R object of "FCquan" or "MCquan" function for the "PTI" study type. *You can directly use the corresponding object stored in R environment, or save it after running the "FCquan" or "MCquan" function and load it when needed.*	/
TIM	vector *(character)*	The method of trajectory inference for the processed data prior to performance evaluation, including "scorpius_distSpear", "scorpius_distPear", "scorpius_distManh", "slingshot_tSNE", "prinCurves_tSNE", "slingshot_PCA", "slingshot_diffMaps" and "prinCurves_diffMaps".	"scorpius_distSpear"
rankingC	vector *(character)*	The copy of the resulting ranking file of "TIassess" function, where a number of workflows that do not require plotting have been deleted.	NULL
Cc_metric	vector *(character)*	The assessing metric under Criterion Cc for the "PTI" study type, including "Spearman correlation" and "Kendall Rank Correlation".	"Spearman correlation"
pathwayhierarchy	vector *(character)*	The absolute filepath of the pathway hierarchy file. *The exampler is provided with the name of "Pathway_Hierarchy.csv" in the folder of "exampler".*	NULL
cores	vector *(numeric)*	The number of CPU cores to be employed for performing parallel computing. *To avoid memory explosion due to parallel computing, the default is the largest integers not greater than half of the number of CPU cores on the current host.*	floor(detectCores()/2)

(func8). exportFCS()

***Description*:** this function exports the quantified CySCP data in the format of FCS and/or expression count matrix (markers in column and events/cells in row). It supports the export of quantification results generated by workflows of top-performance or user interest.

Argument	**Type**	**Description of the Argument and the Allowable Values**	**Default**
data	list	The resulting R object of "FCquan" or "MCquan" function. *You can directly use the corresponding object stored in R environment, or save it after running the "FCquan" or "MCquan" function and load it when needed.*	/

workflow	vector *(character)*	The names of workflows of top-performance or user interest (in the format of "compensationM_transformationM_normalizationM_signalcleanM"). *If NULL, the parameters of "ntop" and "ranking_data" would be used to specify the workflows of top-performance.*	NULL
ntop	vector *(numeric)*	The number of top-performing workflows. *If NULL, the parameter of " workflow" would be used to specify the workflows of top-performance or user interest.*	NULL
ranking_data	vector *(character)*	The absolute filepath of the resulting file (.csv) of the "ranking" function. *This parameter should be used in combination with the parameter of "ntop" to specify the workflows of top-performance.*	NULL
FCS	vector *(logical)*	The logical value specifying whether to export in the format of FCS (TRUE) or not (FALSE). *The parameters of 'FCS' and 'matrix' cannot be set as 'FALSE' at the same time.*	TRUE
matrix	vector *(logical)*	The logical value specifying whether to export in the format of expression count matrix (TRUE) or not (FALSE). *The parameters of 'FCS' and 'matrix' cannot be set as 'FALSE' at the same time.*	FALSE

Table S5. A variety of output files generated by ANPELA 2.0 together with their descriptions.

(func1). ranking()

Name	Brief Description
OUTPUT-ANPELA2023-Overall.Ranking.Data.csv	A csv file containing all information of ranking and the metric value under each criterion.
OUTPUT-ANPELA2023-Overall.Ranking.Figure.pdf	A heatmap illustrating the performance ranking of all workflows based on the metric under each criterion selected by user.

(func2). Classplot()

Name	Brief Description
./OUTPUT-ANPELA2023-Criteria.Ca/class_A_plot-[the name of each workflow].pdf	A pdf file illustrating the ROC curve of each cluster based on each workflow.
./OUTPUT-ANPELA2023-Criteria.Cb/class_B1_plot-[the name of each workflow].pdf	A pdf file showing the cluster distribution of each workflow.
./OUTPUT-ANPELA2023-Criteria.Cb/class_B2_plot-[the name of each workflow].pdf	A pdf file providing the dimension reduction plot colored by sample group and clustering information of each cluster based on each workflow.
./OUTPUT-ANPELA2023-Criteria.Cb/class_B3_plot-[the name of each workflow].pdf	A pdf file depicting the two-way clustering of each cluster based on each workflow.
./OUTPUT-ANPELA2023-Criteria.Cc/class_C1_plot-[the name of each workflow].pdf	A pdf file demonstrating the biomarker overlap Venn diagram selected from each cluster based on each workflow.
./OUTPUT-ANPELA2023-Criteria.Cc/class_C2_plot-[the name of each workflow].pdf	A pdf file providing the volcano plot of each cluster based on each workflow.
./OUTPUT-ANPELA2023-Criteria.Cd/class_D_plot-[the name of each workflow].pdf	A pdf file illustrating the boxplot of the protein expression variations of each workflow.

(func3). TIplot()

Name	Brief Description
./OUTPUT-ANPELA2023-Criteria.Ca/TI_A1_plot-[the name of each workflow].pdf	A pdf file illustrating the trajectory with color-coding cells of each workflow.
./OUTPUT-ANPELA2023-Criteria.Ca/TI_A2_plot-[the name of each workflow].pdf	A pdf file showing the abundances against pseudo-time faceted by real time of each workflow.
./OUTPUT-ANPELA2023-Criteria.Cb/TI_B_plot-[the name of each workflow].pdf	A pdf file providing the contrast of pooled expression variations of each workflow.
./OUTPUT-ANPELA2023-Criteria.Cc/TI_C_plot-[the name of each workflow].pdf	A pdf file depicting the correlation of the inferred pseudo-time between original and partial dataset of each workflow.
./OUTPUT-ANPELA2023-Criteria.Cd/TI_D1_plot-[the name of each workflow].pdf	A pdf file demonstrating the of protein expression plot of each workflow.
./OUTPUT-ANPELA2023-Criteria.Cd/TI_D2_plot-[the name of each workflow].pdf	A pdf file providing the pathway hierarchy correspondence plot of each workflow.

(func4). exportFCS()

Name	Brief Description
./[the name of each workflow]/[the name of each original data file].fcs	A FCS file containing the quantified CySCP data generated by the specified workflow(s).
./[the name of each workflow]/[the name of each original data file].txt	A TXT file containing the quantified CySCP data generated by the specified workflow(s).