CIPHE CMP app tutorial

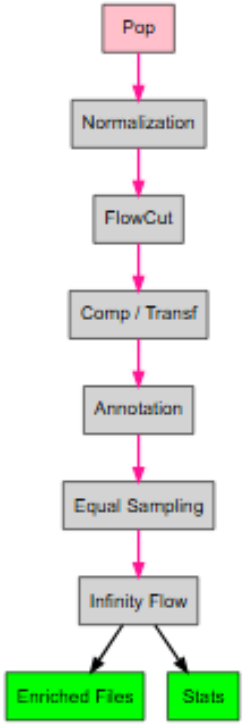
Maelle Marine MONIER
Bioinformatician

Objectives

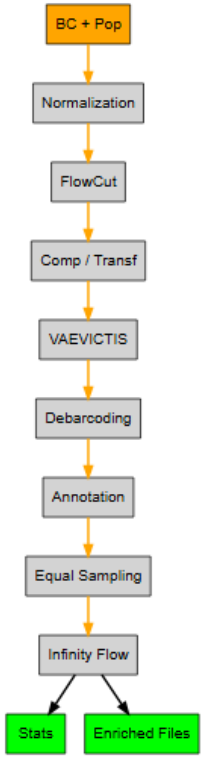
The aim of this application is to interface analysis tools in a workflow for the analysis of CMP (computational cytometry) data. This makes it possible to manage large quantities of files simultaneously.

3 workflows are possible in this app, depending on the project:

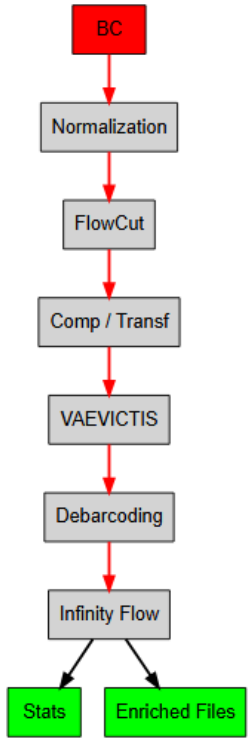
1- Analysis on pop without barcoding



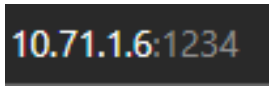
2- Analysis on pop + barcoding



3- Analysis on one population with barcodes



App accessibility



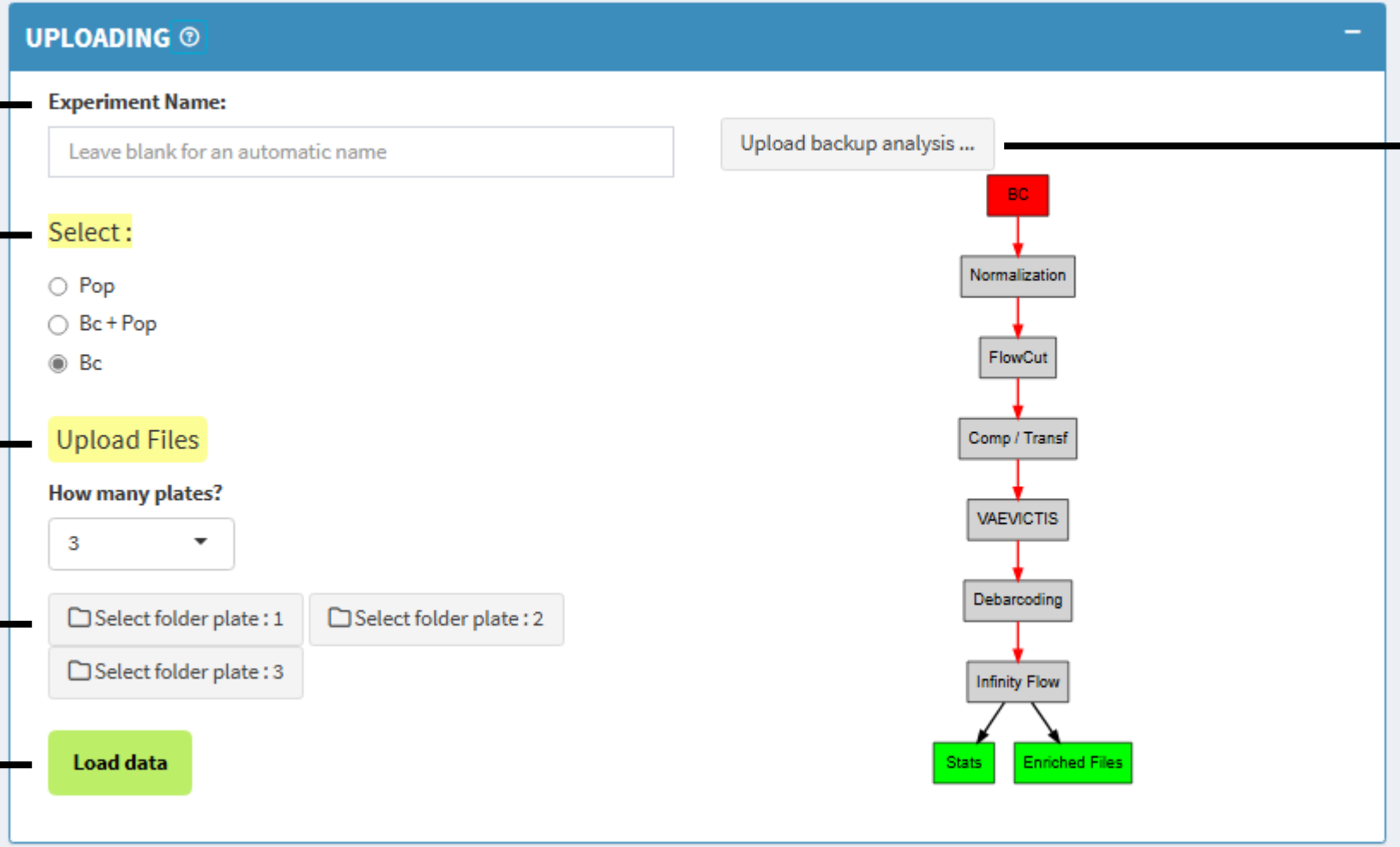
1 - To access to the app, go to your favorite browser and tap
10.71.1.6:1234
OR go to GoT (10,71,1,22)

2 – In WINSCP, create a folder for your experiment here:
/mnt/md0/CMP/input

And put your files in it (one folder per plate)

CIPHE CMP app:
DATA MANAGMENT

DATA MANAGMENT



Name of your experiment

Select your experiment (Bc = barcode)

Indicate your plate number

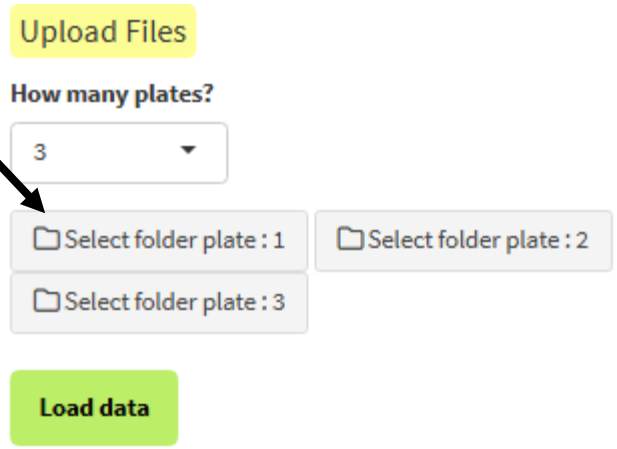
Choose the path to your FCS files for each plate

Load the FCS files

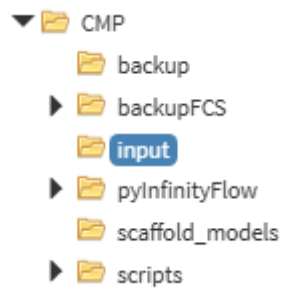
Upload

STEP BY STEP : load your data

1- For each plate please click on the button

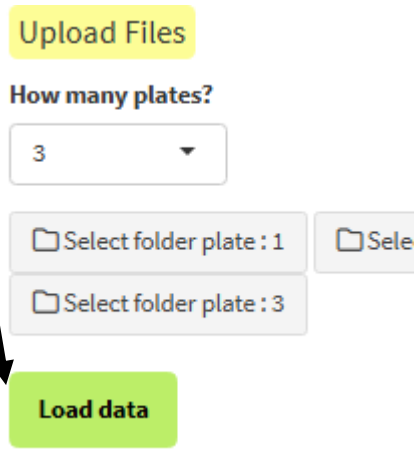


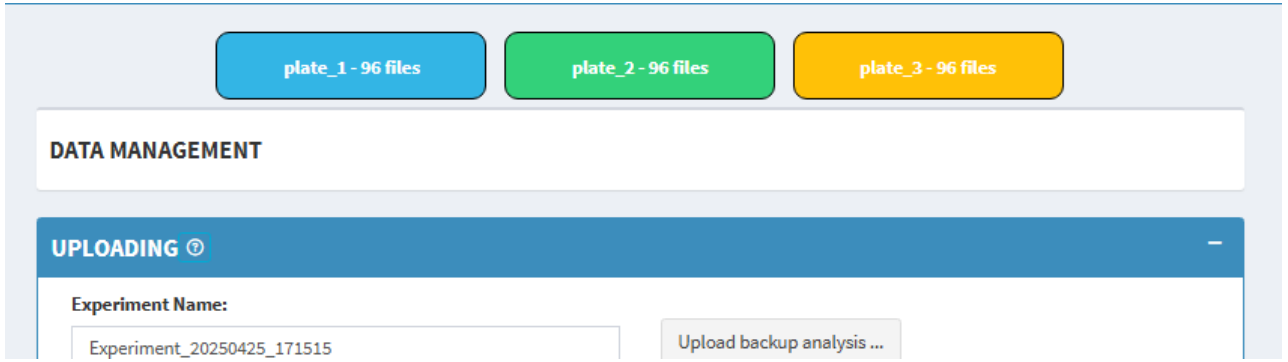
2- Click on path to your FCS for the selected plate



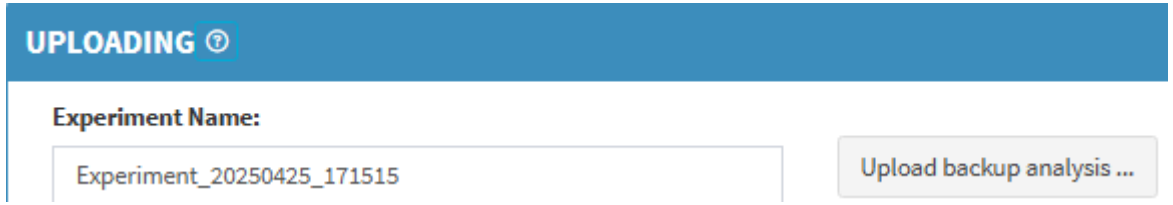
3- Repeat these steps for the other plates

4- Click on Load data



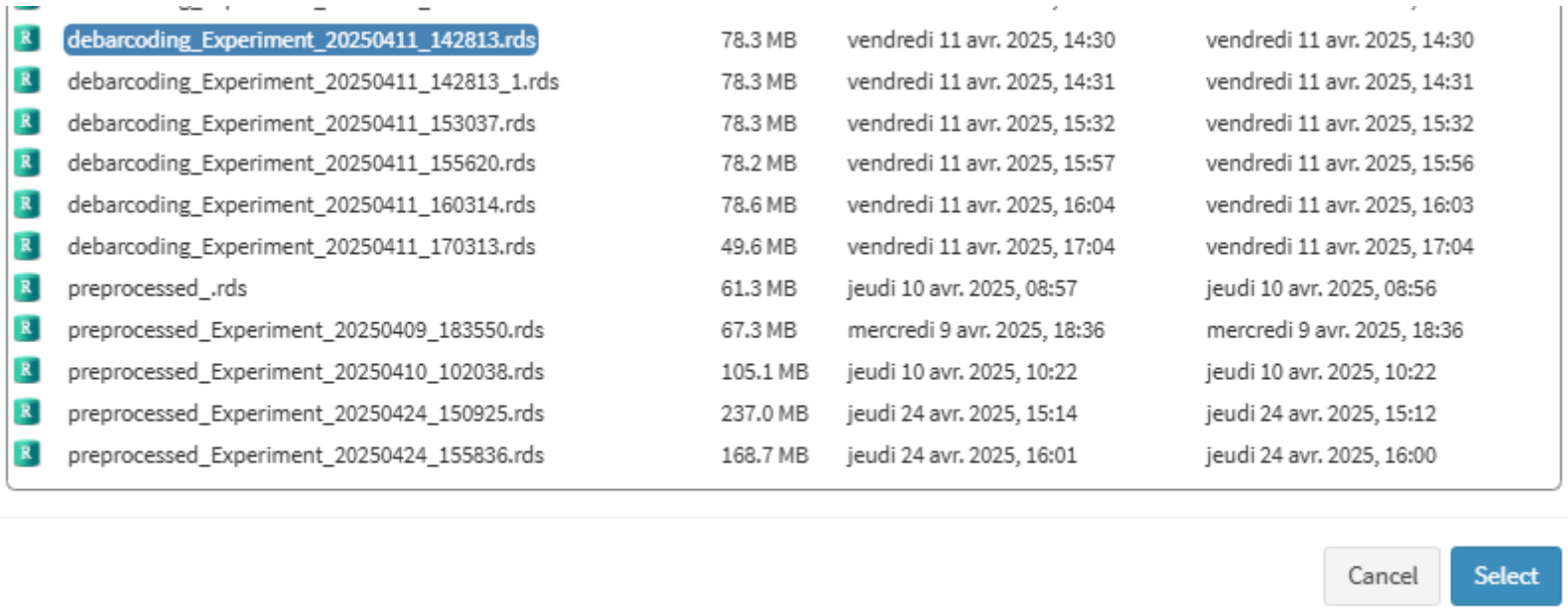


STEP BY STEP : load your data



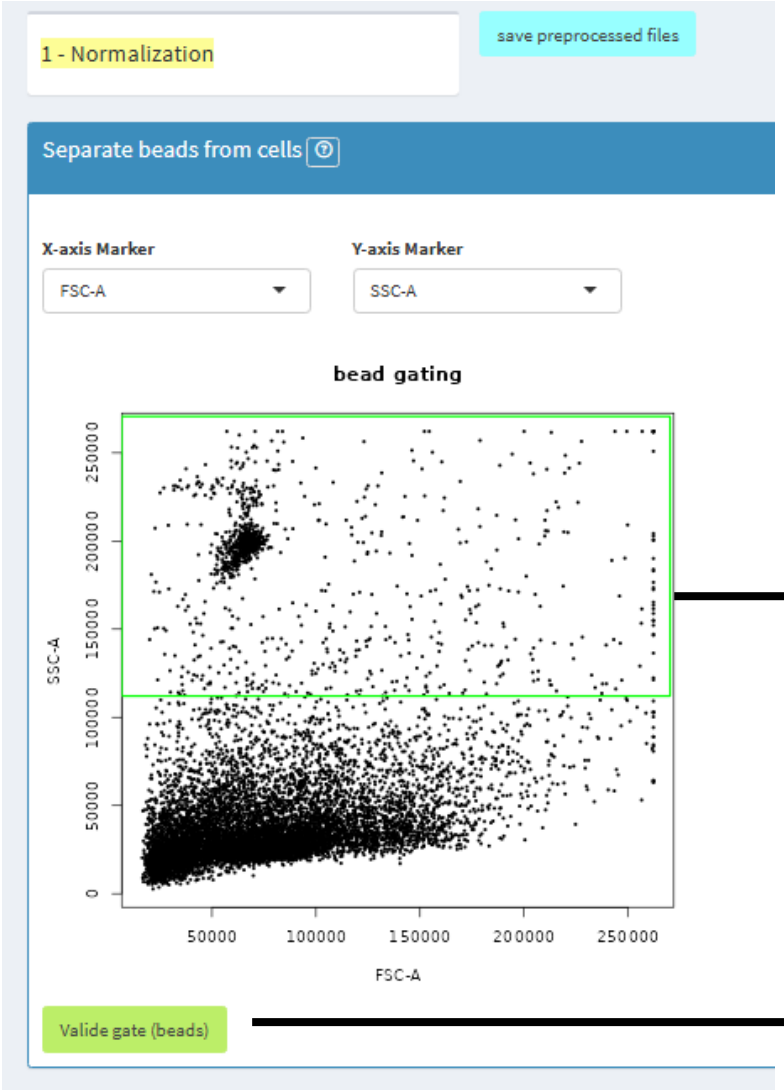
RDS backup:

Click on **Upload backup analysis**.
Select the RDS project file you previously saved.



CIPHE CMP app:
PREPROCESSING

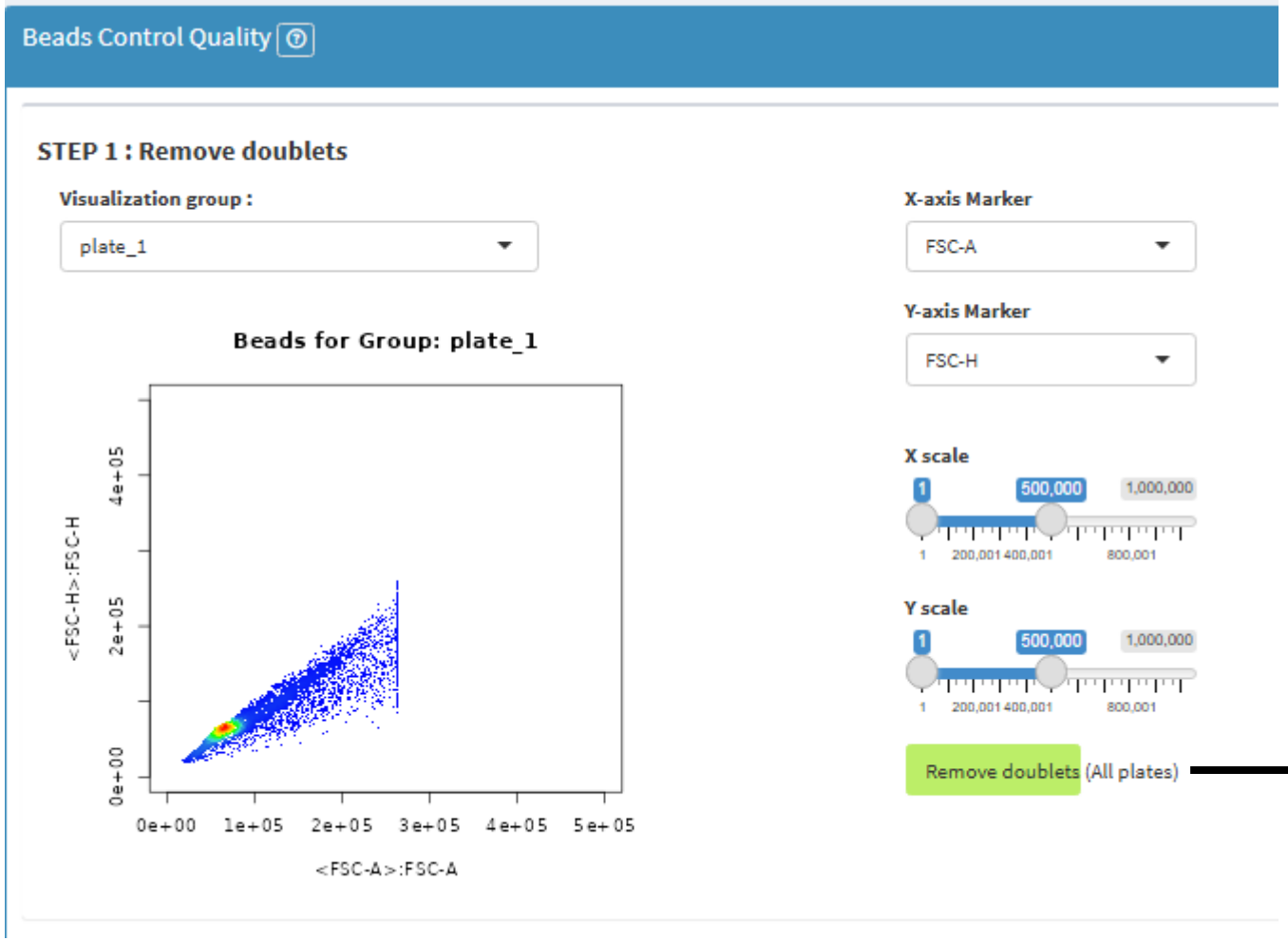
1 – NORMALIZATION BETWEEN PLATES



With your mouse, select beads

Then, click on validate gate. This step will extract beads for the normalization and also remove all events in the selections from your files

1 – NORMALIZATION BETWEEN PLATES



Bead gating step 1 :
Click to remove doublets from your beads

1 – NORMALIZATION BETWEEN PLATES

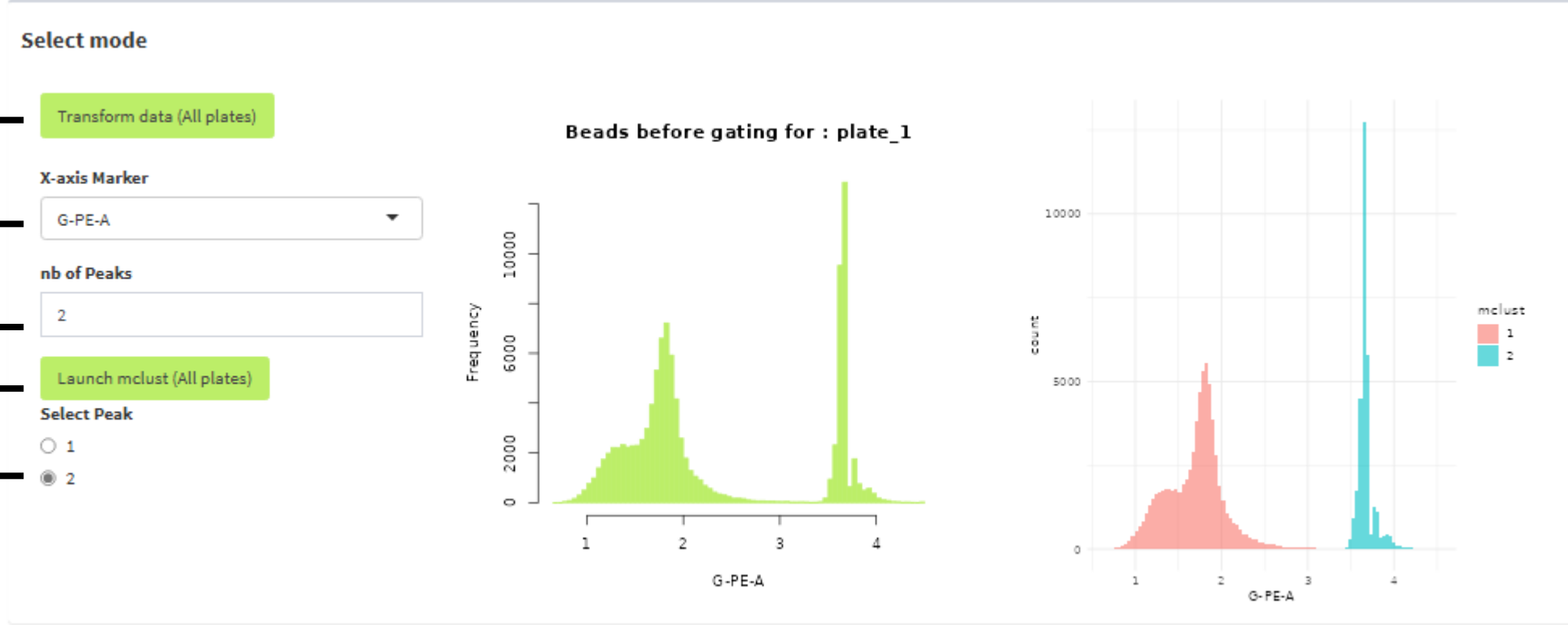
1- Transform beads with logicle(500) (just for better visualization)

2- Choose marker with the better distinction between peaks

3- Indicate the number of bead peaks

4- Launch clusterint algorithm

5- Choose the bead peak that you want to use for the normalization



1 – NORMALIZATION BETWEEN PLATES

(optionnal) STEP 3 : Re-gate

Select group

plate_1

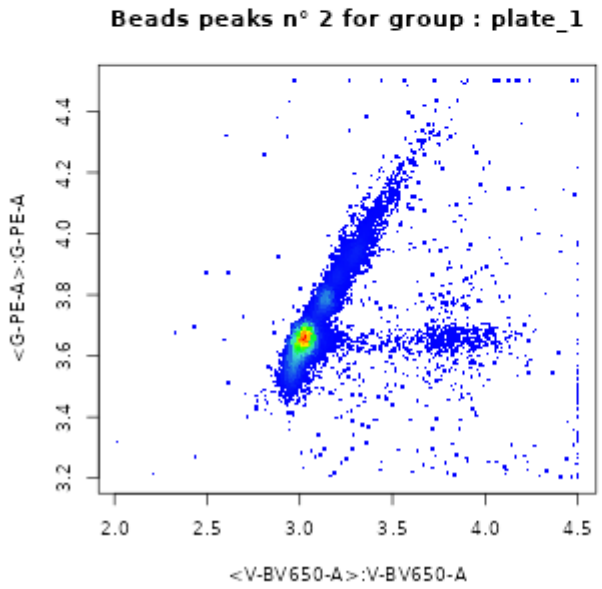
X-axis Marker

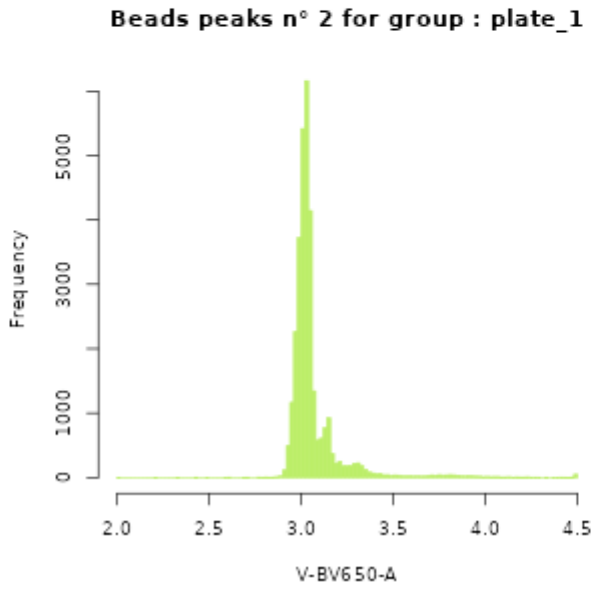
V-BV650-A

Y-axis Marker

G-PE-A

Reset Gate





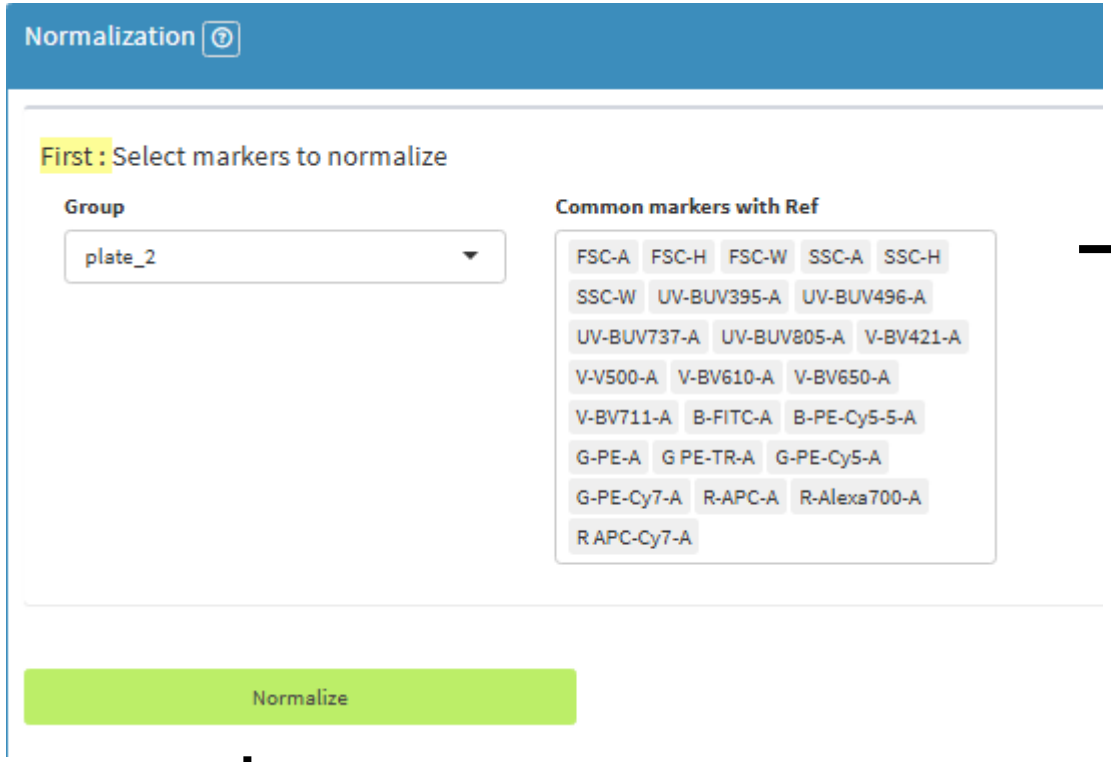
For each plate, perform manual gating to select only beads and not debris

↓ And after

unTransform data (All plates) → Click on untransform beads

Validate beads (All plates) → And finally, click on validate beads

1 – NORMALIZATION BETWEEN PLATES



→ Select markers to normalize for each Group

↓ Then, click on Normalize

Normalization outputs

Harmonization table for :

plate_2

Show 10 entries Search:

	Factor
FSC-A	0.9856
FSC-H	0.9896
FSC-W	0.9971
SSC-A	0.9999
SSC-H	1.0016
SSC-W	0.9983
UV-BUV395-A	1.5335
UV-BUV496-A	0.5629
UV-BUV737-A	3.1914
UV-BUV805-A	4.2899

Showing 1 to 10 of 24 entries Previous 1 2 3 Next

Choose

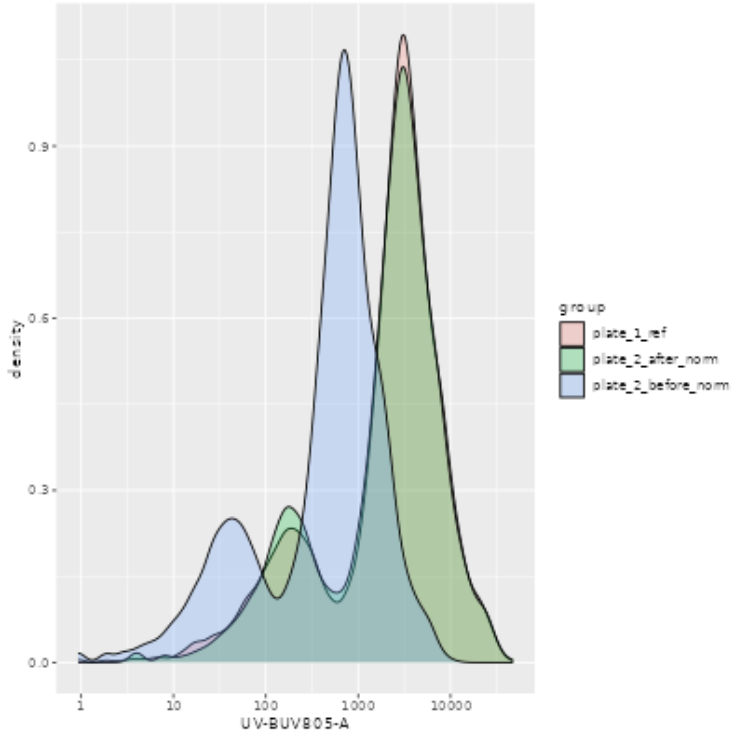
rawFCS

X-axis Marker

UV-BUV805-A

Group

plate_1_ref
plate_2_before_norm
plate_2_after_norm



Enter ZIP file name (without extension):

NormalizedFCS

Download normalized FCS

Download harmonized Table

2 – FLOWQC

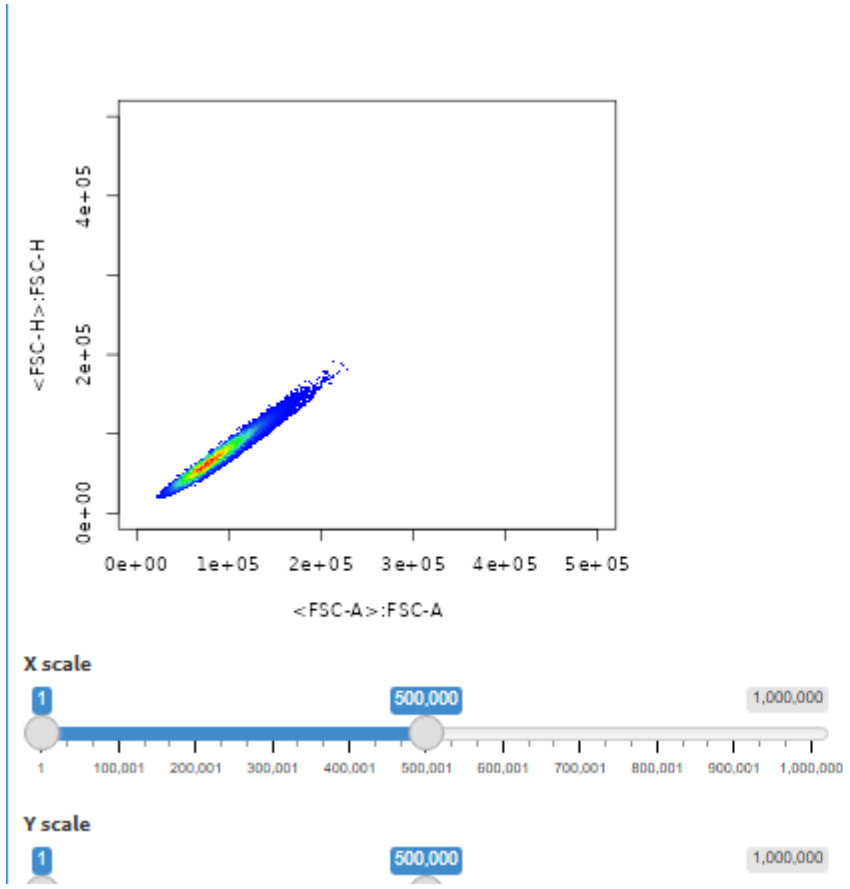
This section is designed to clean your FCS files by removing technical artifacts and unwanted events.

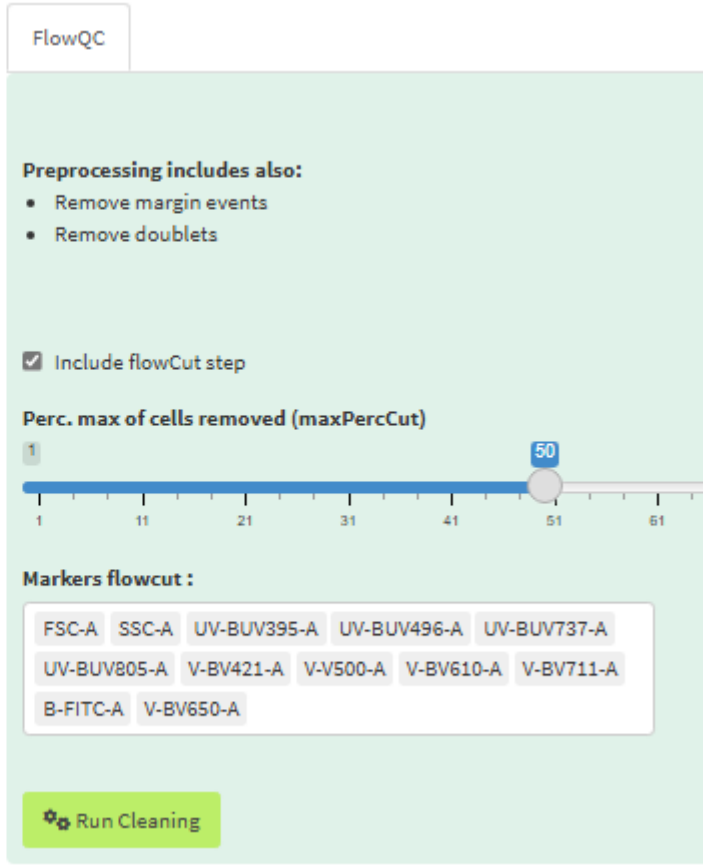
Remove margin events: These are extreme events that fall outside the detector range and may distort analysis.

Remove doublets: These are overlapping events (e.g., two cells passing together) that should be excluded from single-cell analysis.

Use FlowCut: FlowCut is an automated quality control algorithm that detects and removes abnormal regions in flow cytometry data over time. It uses statistical and signal drift detection to identify unstable acquisition periods, improving data consistency.

→ Configure FlowCut thresholds and select relevant markers before launching the cleaning process on all files.





flowCut output

Results QC :

plate_1 | plate_2

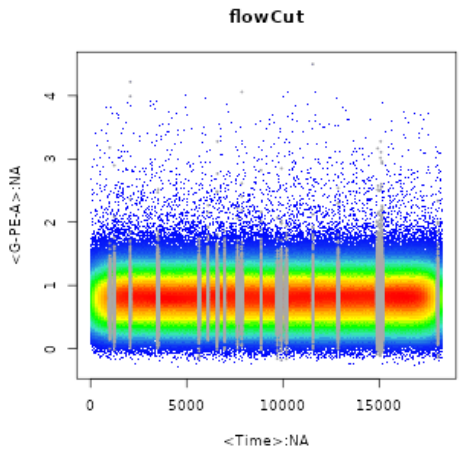
Show 10 entries Search:

plate_2_Specimen_001_MIX+Billes-phoenix_003.fcs	Perc.Removed
plate_2_Specimen_001_MIX+Billes-phoenix_003.fcs	2.8262

Showing 1 to 1 of 1 entries Previous 1 Next

	[,1]
Is it monotonically increasing in time	"T"
Largest continuous jump	"0.828"
Continuous - Pass	"F"
Mean of % of range of means divided by range of data	"0.153"
Mean of % - Pass	"F"
Max of % of range of means divided by range of data	"0.871"
Max of % - Pass	"F"
Has a low density section been removed	"F"
% of low density removed	"0"
How many segments have been removed	"41"
% of events removed from segments removed	"2.8262"
Worst channel	"G-PE-A"
% of events removed	"2.8262"
FileID	"2025_04_28__09_39_47"
Type of Gating	"AllCutMin"
Was the file run twice	"Yes"
Has the file passed	"F"

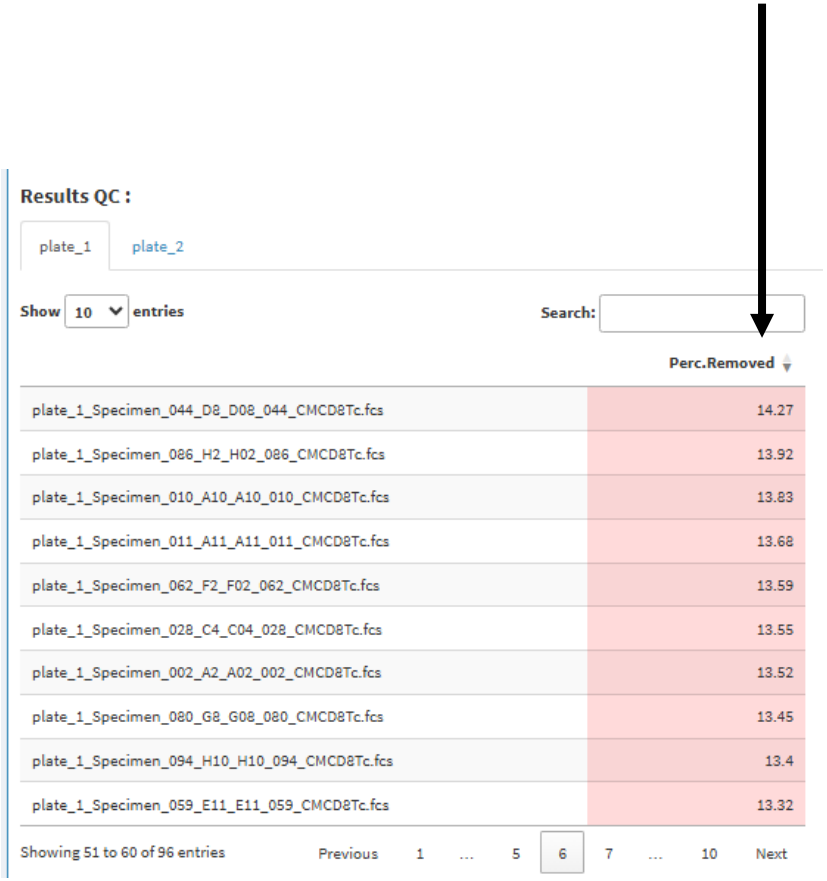
RemoveQC (Selected file) | RemoveQC (Selected group) | ValidateQc (All Files)



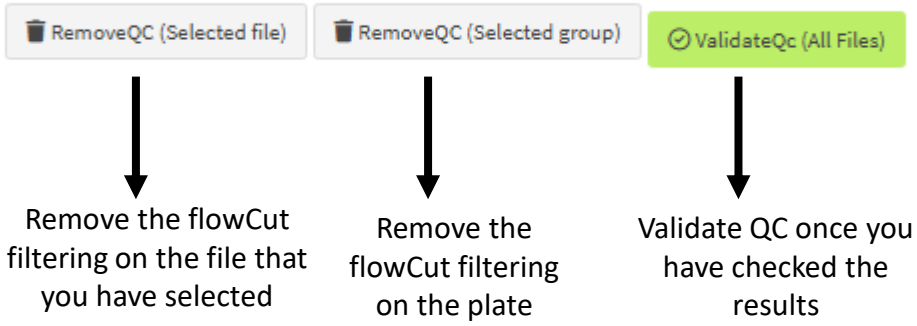
For each plate, you have a table with the file name, and the associated percentage of deleted cells. For more information on a particular file, click on the file in the table.

flowCut outputs

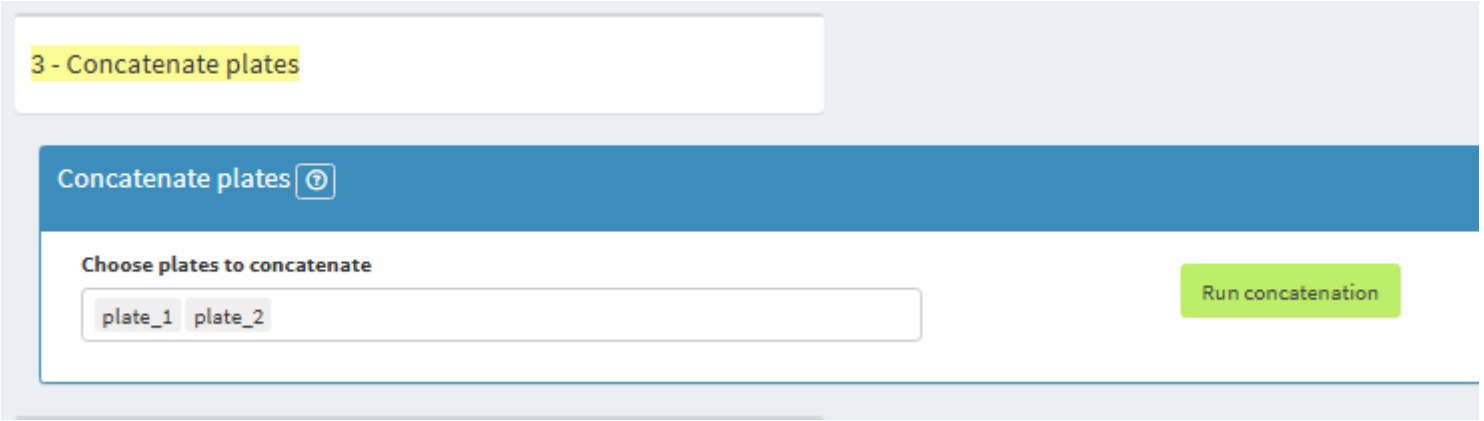
Click on Perc.Removed to display the files in ascending or descending order.



SAVE FLOWCUT FILTERING :



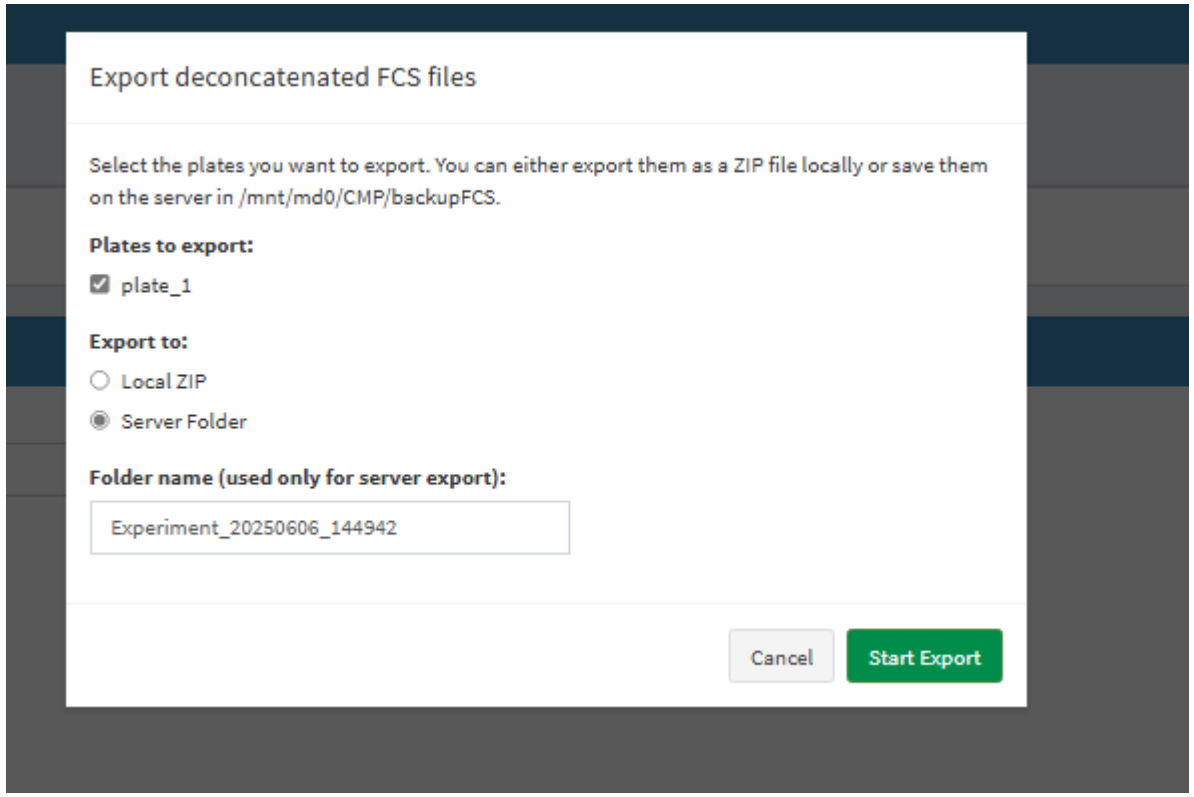
3- Concatenate plates



This section allows you to concatenate plates. The data parameters will be those present in the first plate you selected.

! Repeat this action as many times as you have plates to concatenate. !

3- Concatenate plates



After concatenation, export the plates containing the files (e.g. if you concatenated Plate 1 and Plate 2, export Plate 1).

Then refresh the application and re-import the files.

4-Compensation and transformation

Apply comp

Upload csv file, divided in two column (Fluo and Arg) delimited by a comma

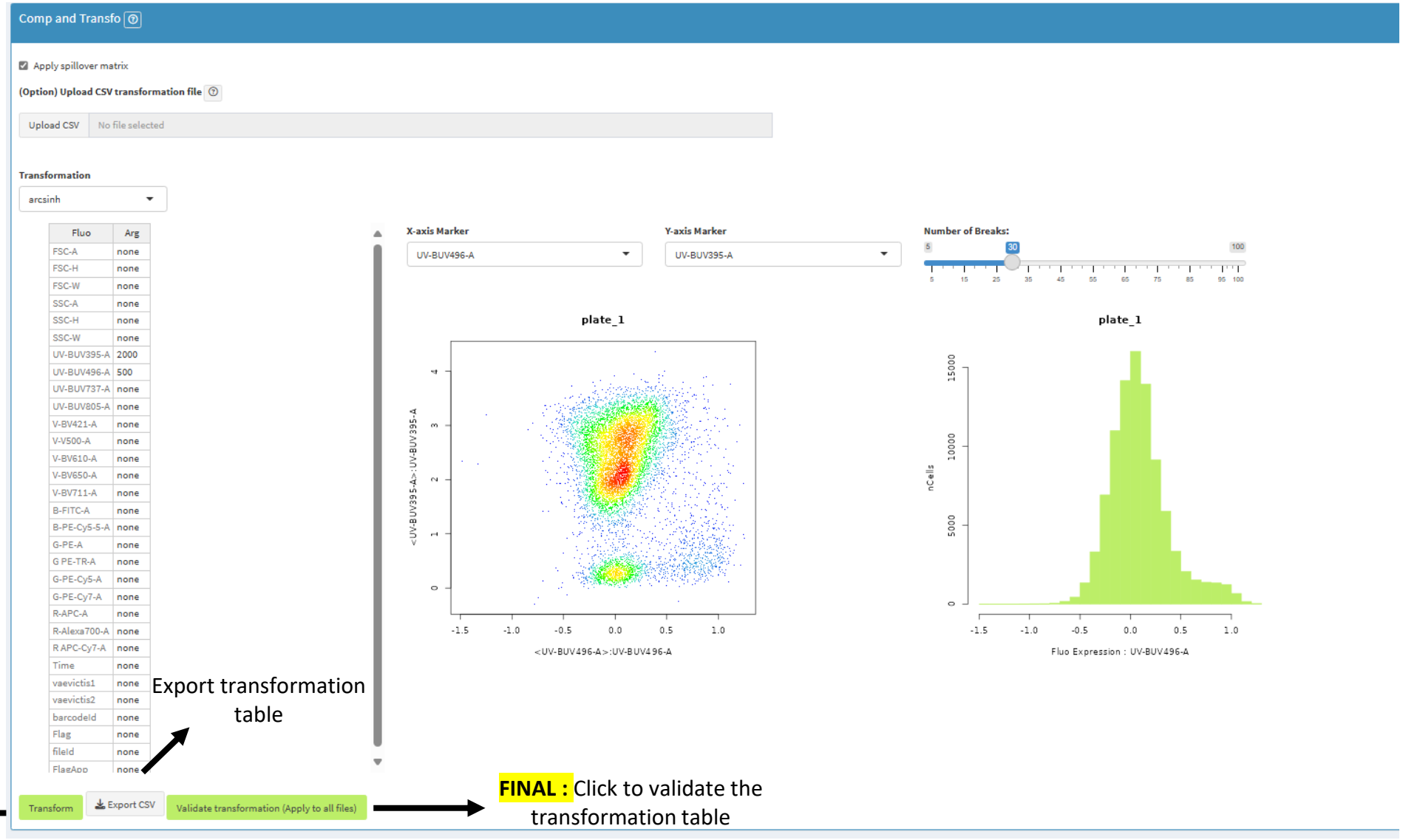
Choose transfo : arcsinh, logicle

Change cofactor

Export transformation table

Apply transform to visualize your data transformation

FINAL : Click to validate the transformation table



CIPHE CMP app: **DEBARCODING**

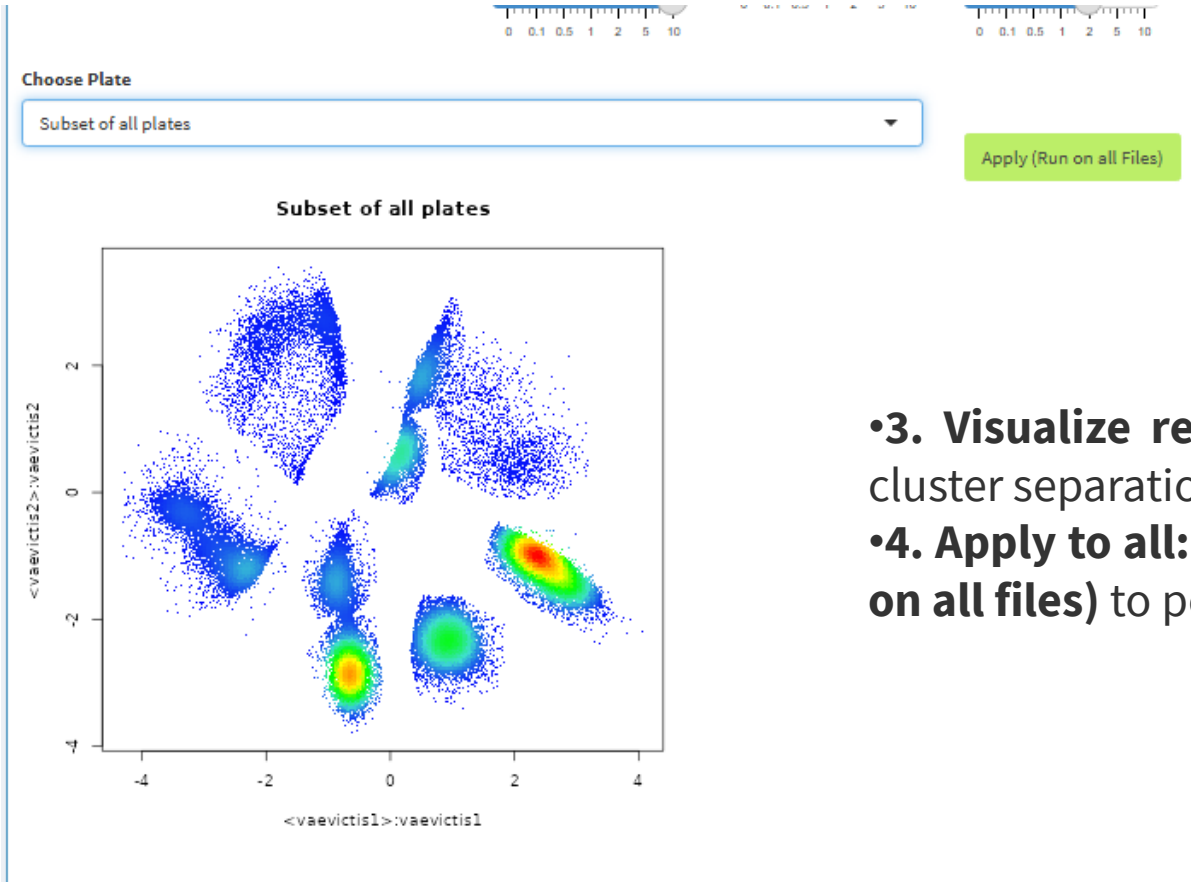
1- Run vaevictis

Run vaevictis on subset of each file and each plate

VAEVICTIS RUN:

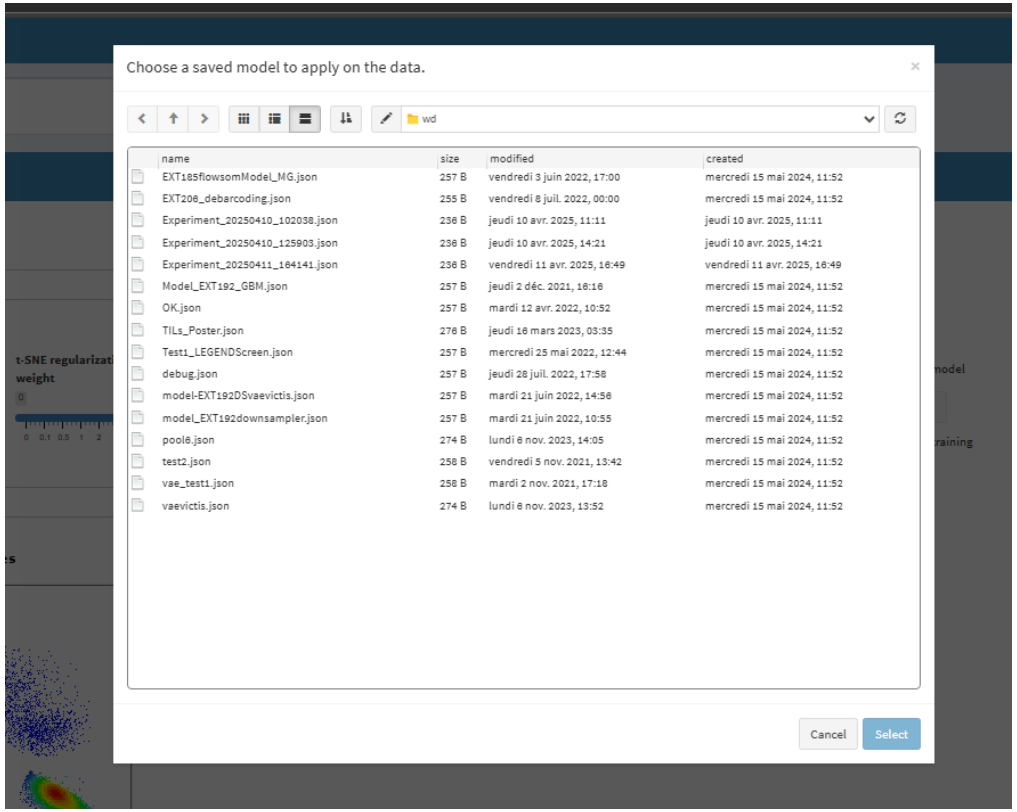
- **1. Marker selection:** Choose the markers you want to use for clustering. These should be relevant for distinguishing cell populations.
- **2. Run on subset:** Click **Run** to apply VAEVICTIS on a subset of all your files (combined from all plates).

1- Run vaevictis



- **3. Visualize results:** Review the plots generated to assess cluster separation and quality.
- **4. Apply to all:** If satisfied with the preview, click **Apply (Run on all files)** to perform the full clustering process.

1- Run vaevictis

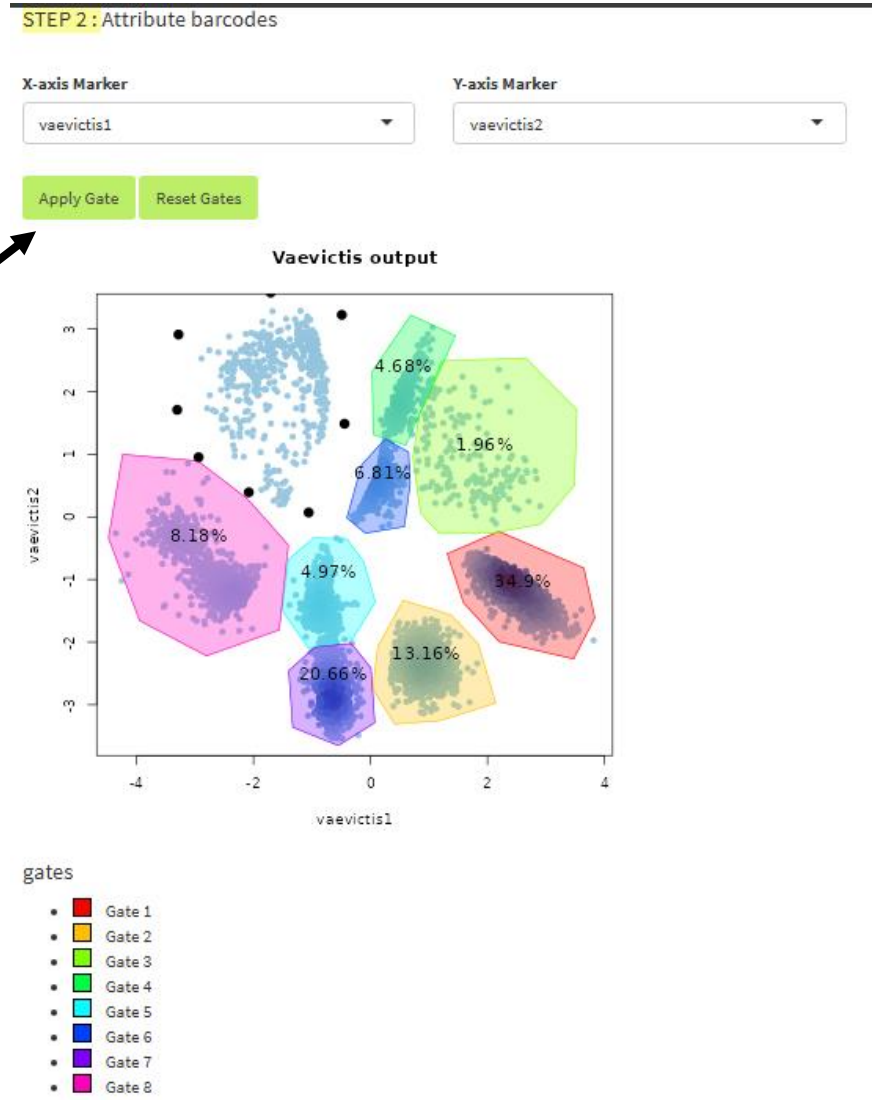


Load a saved model

Apply (Run on all Files)

1- Run vaevictis

Manual gating : just click on the plot to gate barcode, and then, click on apply Gate

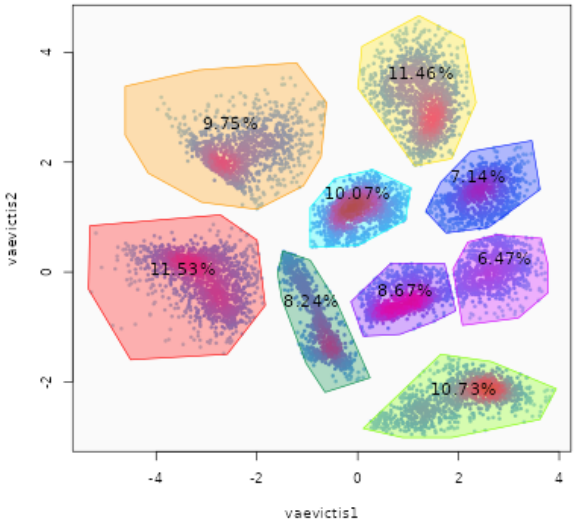


2- debarcoding

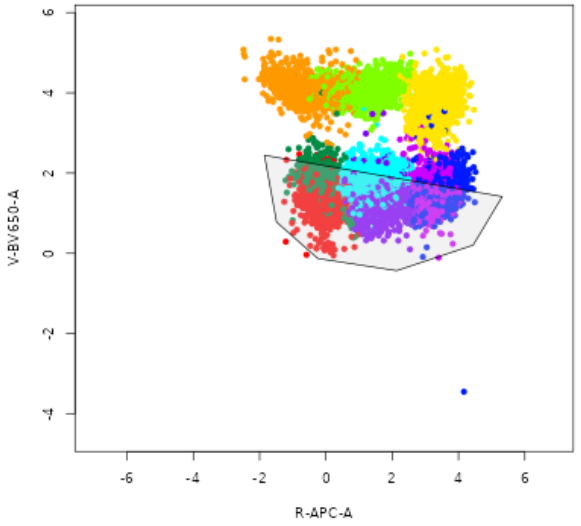
X-axis Marker: FSC-A | Y-axis Marker: FSC-A | X-axis Marker (gate): R-APC-A | Y-axis Marker (gate): V-BV650-A | X-axis Marker (gate 2): R-APC-A | Y-axis Marker (gate 2): UV-BUV395-A

Apply Gate | Reset Gates | Apply Gate | Reset Gates

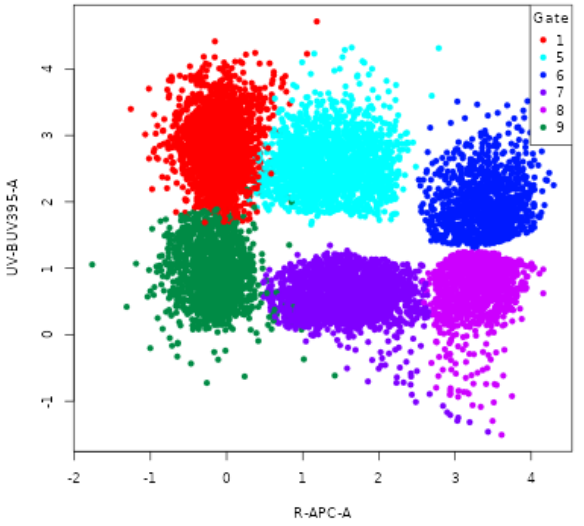
Vaevictis output



Gate Visualization



Gate 2 Visualization



gates

- Gate 1
- Gate 2
- Gate 3
- Gate 4
- Gate 5
- Gate 6
- Gate 7
- Gate 8
- Gate 9

Select

- [x] Gate 1
- [x] Gate 2
- [x] Gate 3
- [x] Gate 4
- [x] Gate 5
- [x] Gate 6
- [x] Gate 7
- [x] Gate 8
- [x] Gate 9

STEP 3 : Reattribute id to each gate (ONLY NUMERIC id)

Visualize each gate/barcode in a second plot. Select gate that you want to see in the plot

		barcode_label
6	7	CC60
7	4	CC8
8	8	CC71

Export table

validate barcoding (All files)

Information:

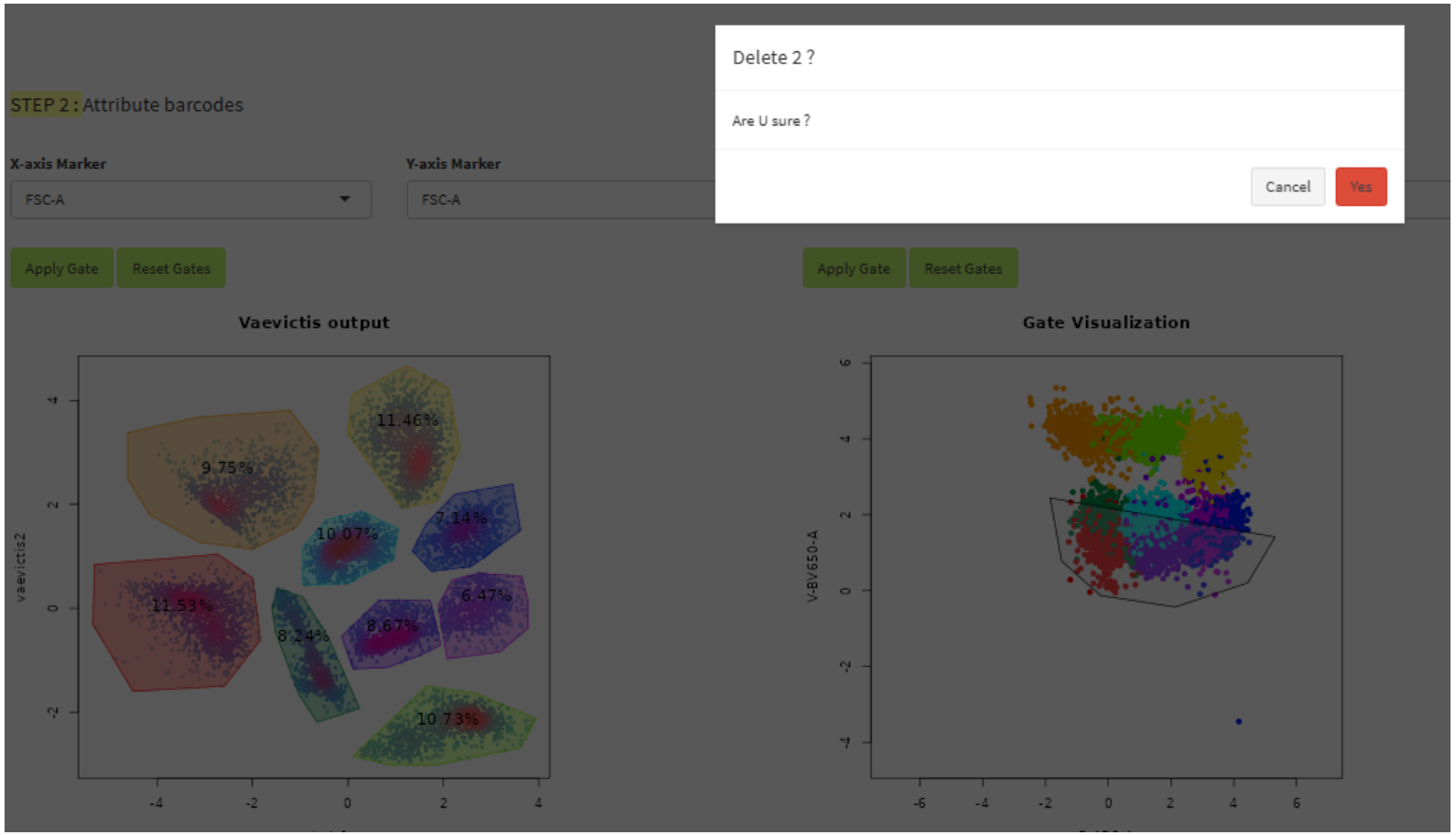
The column **barcode_id** refers to the NUMERICAL values that will b the FCS files to represent barcodes. FCS files can only store nume which is why this column is required.

The column **barcode_name** will not be stored in the FCS files but s human-readable label to help match the **barcode_id** to a corr phenotype or condition. You can use any text here.

You can also export or import this table to preserve and reuse you mappings in future sessions.

2- debarcoding

To delete just one gate, click on the gate



2- debarcoding

STEP 3 : Reattribute id to each gate (ONLY NUMERIC id)

gate	barcode_id	barcode_label
1	1	B6J
2	3	CC80
3	9	CC16
4	6	CC10
5	5	CC12
6	7	CC60
7	4	CC8
8	8	CC71
9	2	CC3

Export table

validate barcoding (All files)

Information:

The column **barcode_id** refers to the NUMERICAL values that will be added to the FCS files to represent barcodes. FCS files can only store numeric values, which is why this column is required.

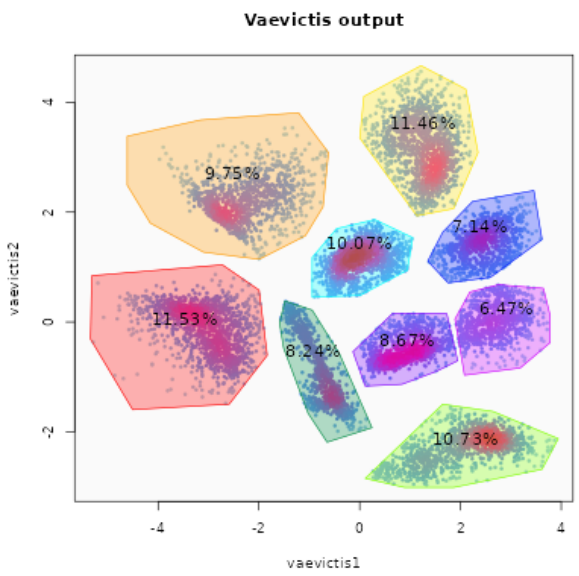
The column **barcode_name** will not be stored in the FCS files but serves as a human-readable label to help match the **barcode_id** to a corresponding phenotype or condition. You can use any text here.

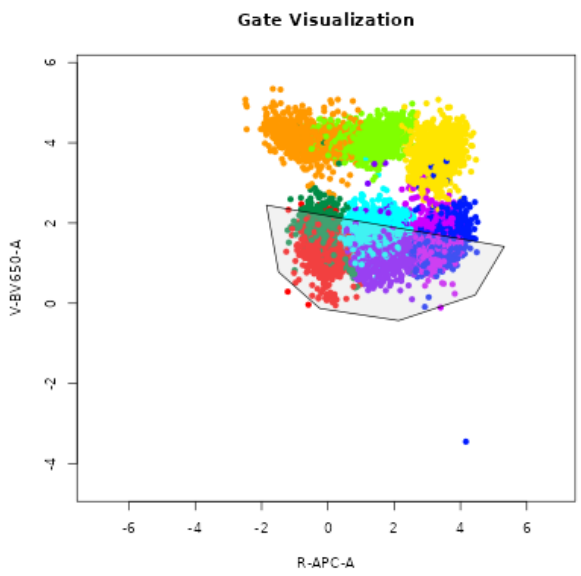
You can also export or import this table to preserve and reuse your barcode mappings in future sessions.

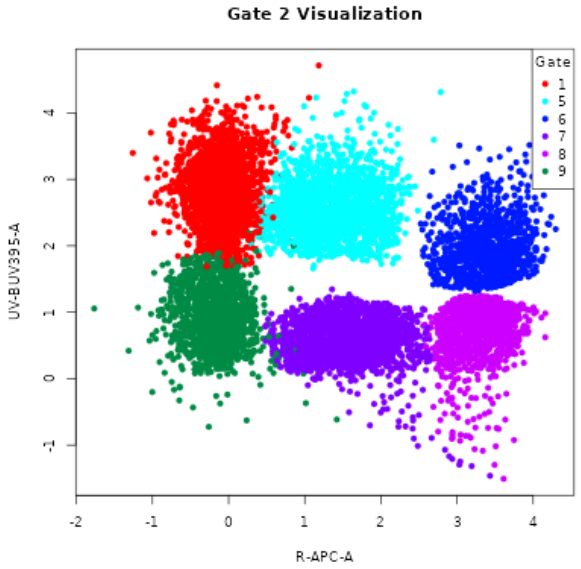
Once the table is debarcoded, click on validate barcode so that it enriches all your files with barcodes (all cells outside the gates will be removed from the files)

X-axis Marker: FSC-A | Y-axis Marker: FSC-A | Apply Gate | Reset Gates

X-axis Marker (gate): R-APC-A | Y-axis Marker (gate): V-BV650-A | X-axis Marker (gate 2): R-APC-A | Y-axis Marker (gate 2): UV-BUV395-A | Apply Gate | Reset Gates







Important information: Only the gates on the left-hand plot count; the gate on the middle plot is for visualization purposes only.

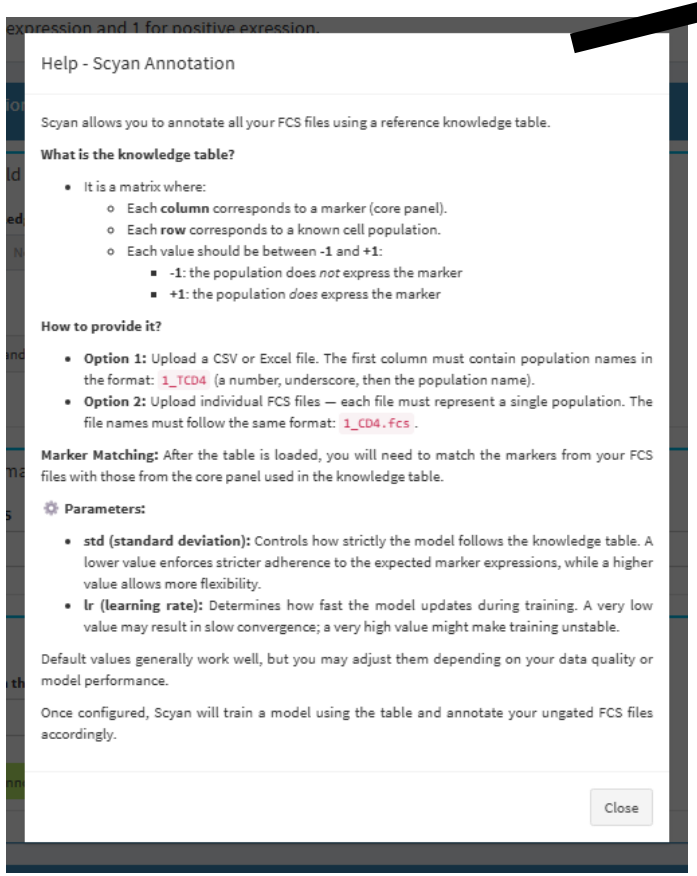
CIPHE CMP app: **ANNOTATION**

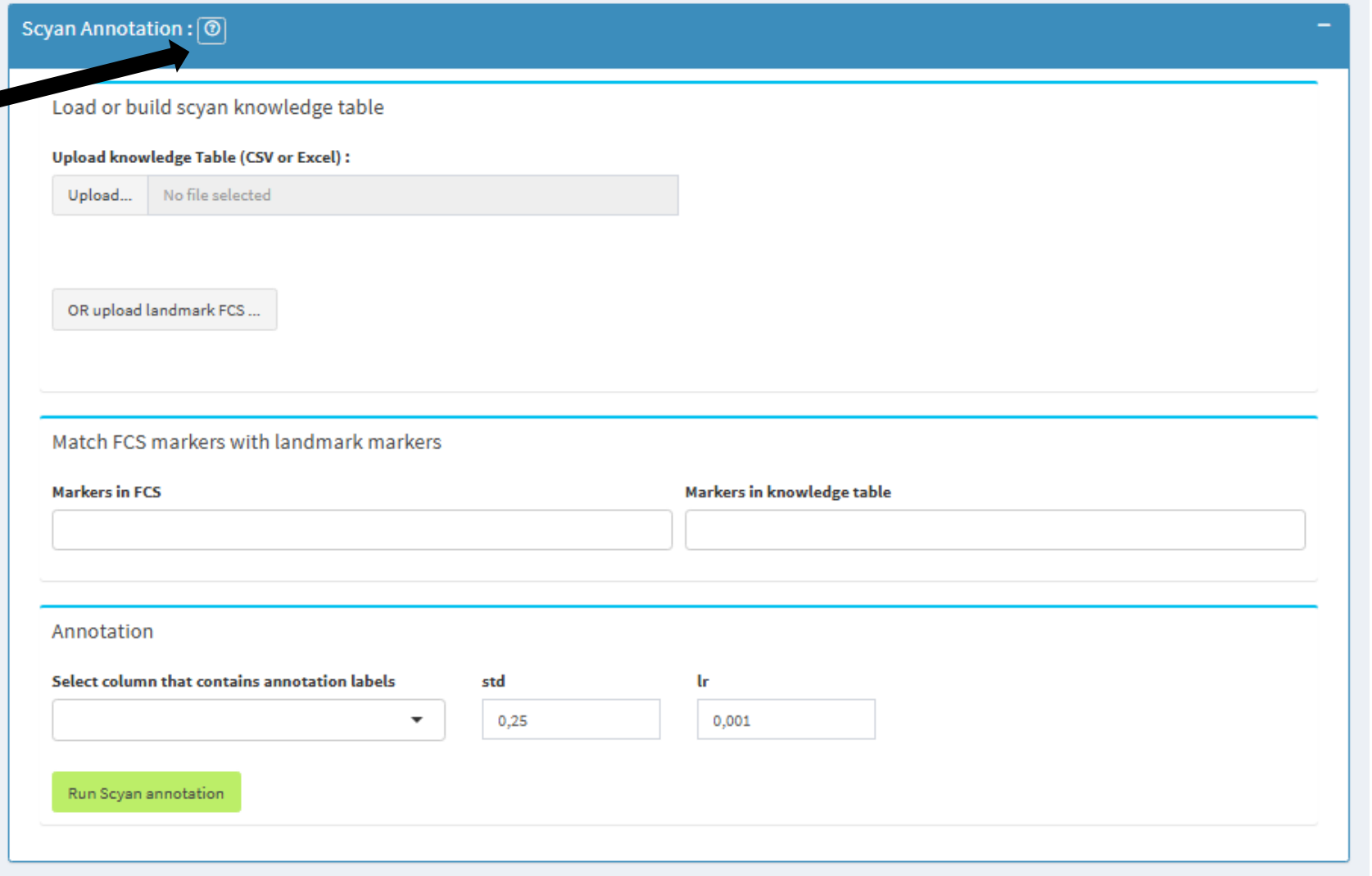
ANNOTATION

This section allows you to annotate a subset of your dataset based on reference files. You can choose between two methods: Scyan or Scaffold. Both methods apply annotations to a subset of the data. Once the annotation is done, scroll down to the 'QC Annotation' section to inspect the results using UMAP and scatter plots. If you are satisfied with the results, click on 'Run on all files' to apply the annotation to all FCS files across your plates.

SCYAN ANNOTATION

As with all parts, click on the help button for information on this part.





SCYAN ANNOTATION

What is the knowledge table?

- It is a matrix where:
  - Each **row** corresponds to a known cell population.
  - Each value should be between **-1** and **+1**:
    - **-1**: the population does *not* express the marker
    - **+1**: the population *does* express the marker

How to provide it?

- **Option 1:** Upload a CSV or Excel file. The first column must contain population names in the format: 1_TCD4 (a number, underscore, then the population name).
- **Option 2:** Upload individual FCS files — each file must represent a single population. The file names must follow the same format: 1_CD4.fcs.

SCYAN ANNOTATION

Marker Matching: After the table is loaded, you will need to match the markers from your FCS files with those from the core panel used in the knowledge table.

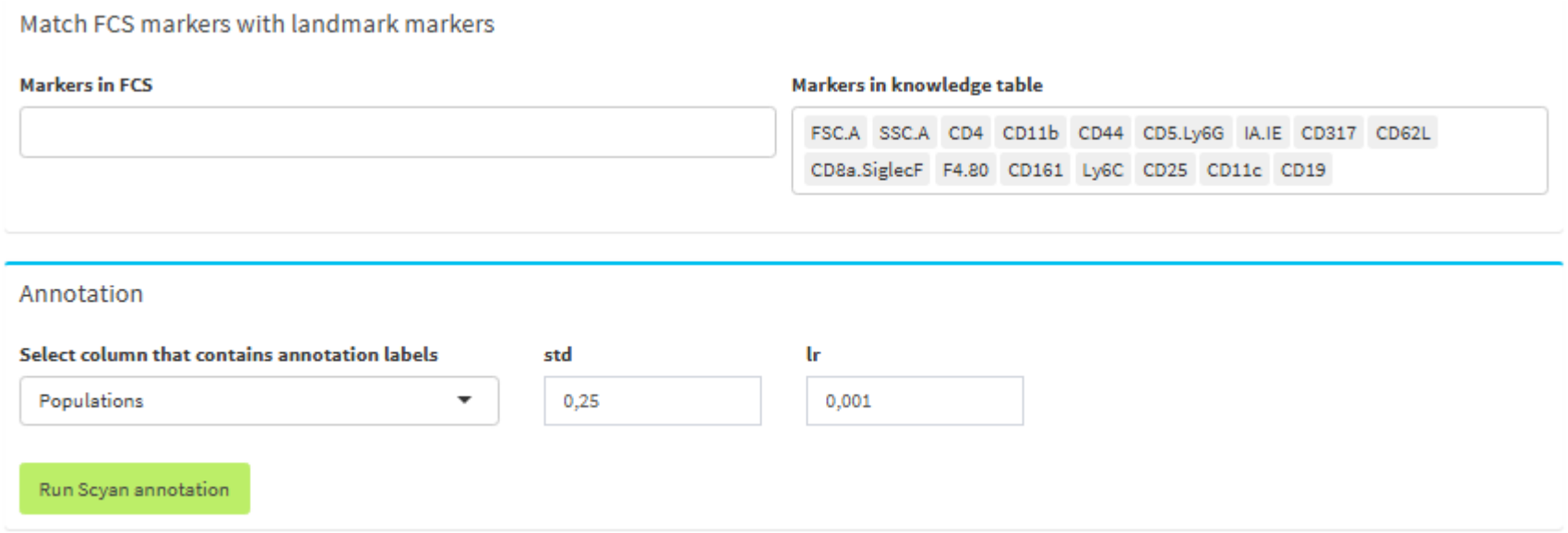
Parameters:

•**std (standard deviation):** Controls how strictly the model follows the knowledge table. A lower value enforces stricter adherence to the expected marker expressions, while a higher value allows more flexibility.

•**lr (learning rate):** Determines how fast the model updates during training. A very low value may result in slow convergence; a very high value might make training unstable.

Default values generally work well, but you may adjust them depending on your data quality or model performance.

Once configured, Scyan will train a model using the table and annotate your ungated FCS files accordingly.



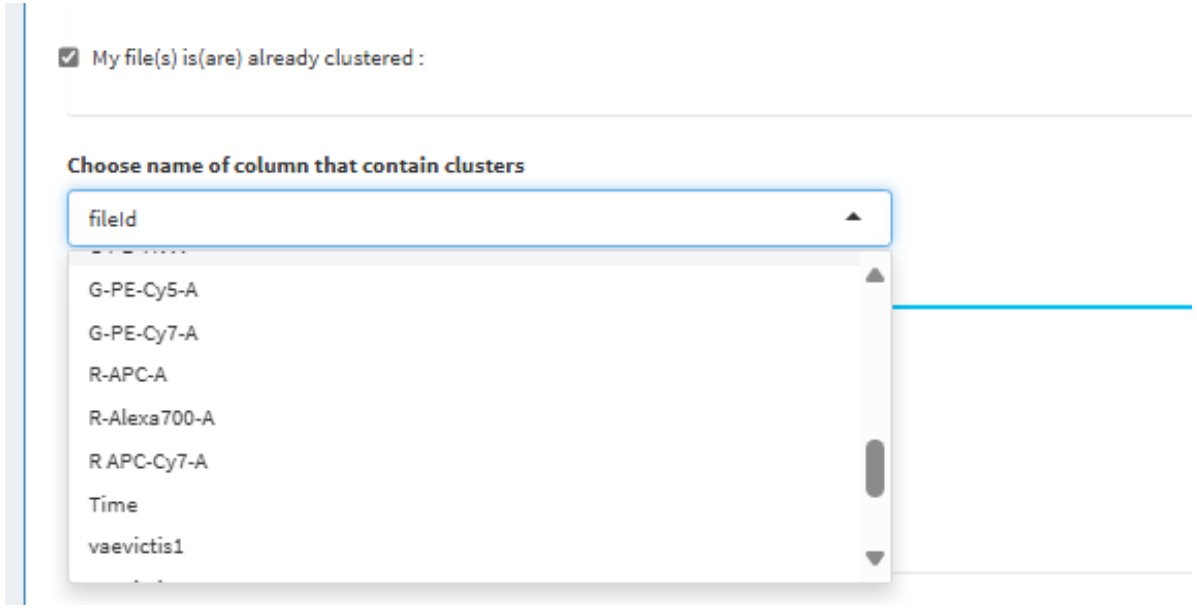
SCAFFOLD ANNOTATION

Step 1 – Clustering:
By default, the clustering algorithm used is **CLARA**.
Select the markers you want to use for clustering and click **Run CLARA clustering**.

SCAFFOLD ANNOTATION

Step 1 – Clustering:

If your file is already clustered, check the box **“My files are already clustered”** and select the column that contains the cluster labels.



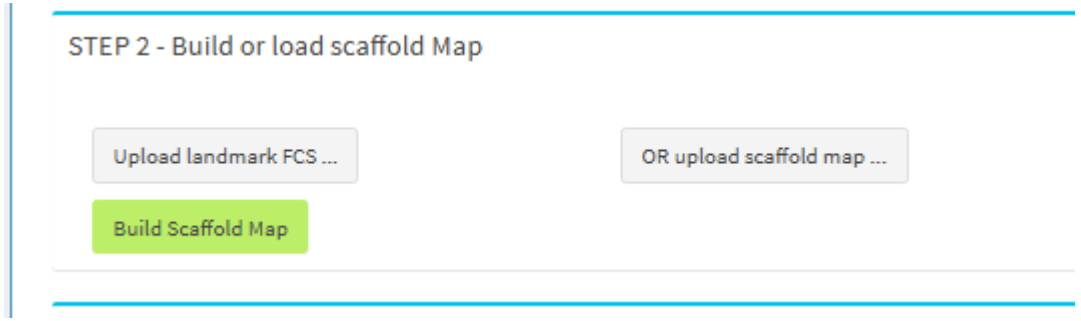
SCAFFOLD ANNOTATION

Step 2 – Build or Load Scaffold Map:

•You can build a Scaffold reference map from annotated FCS files.

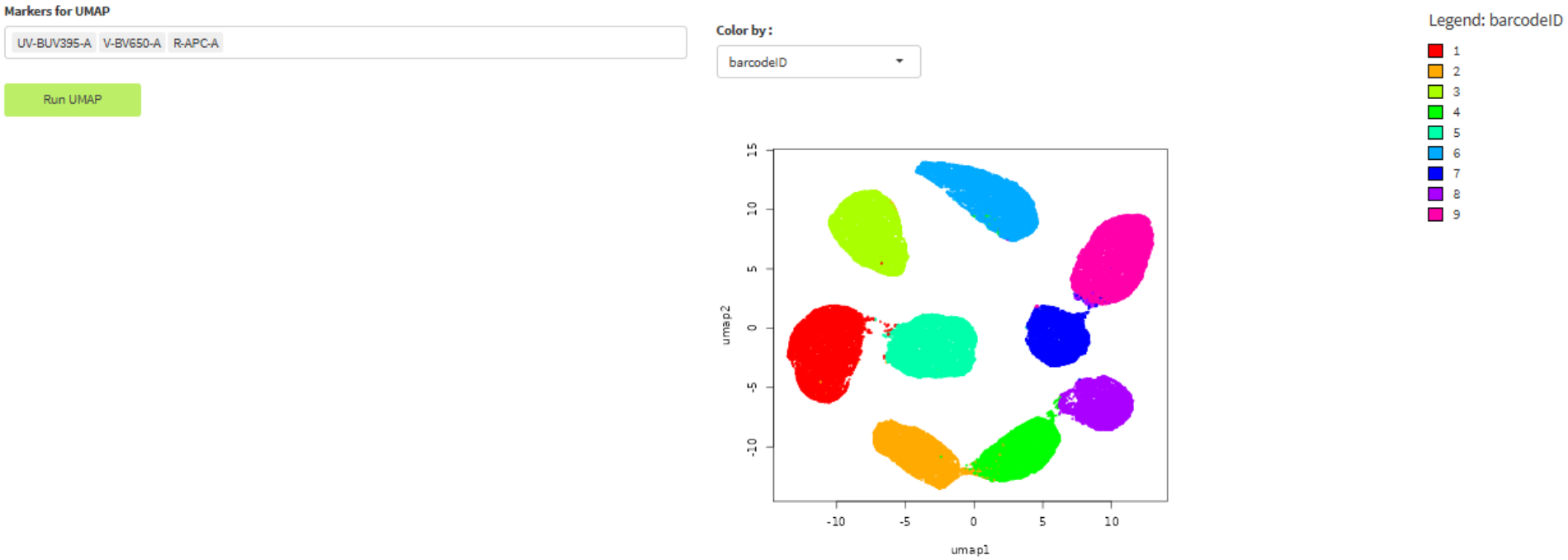
•Each FCS file must contain a single cell population, and the filename should follow the format: 1_TCD4.fcs (ID number + underscore + population name).

•Alternatively, you can upload a previously built Scaffold map if you already have one.

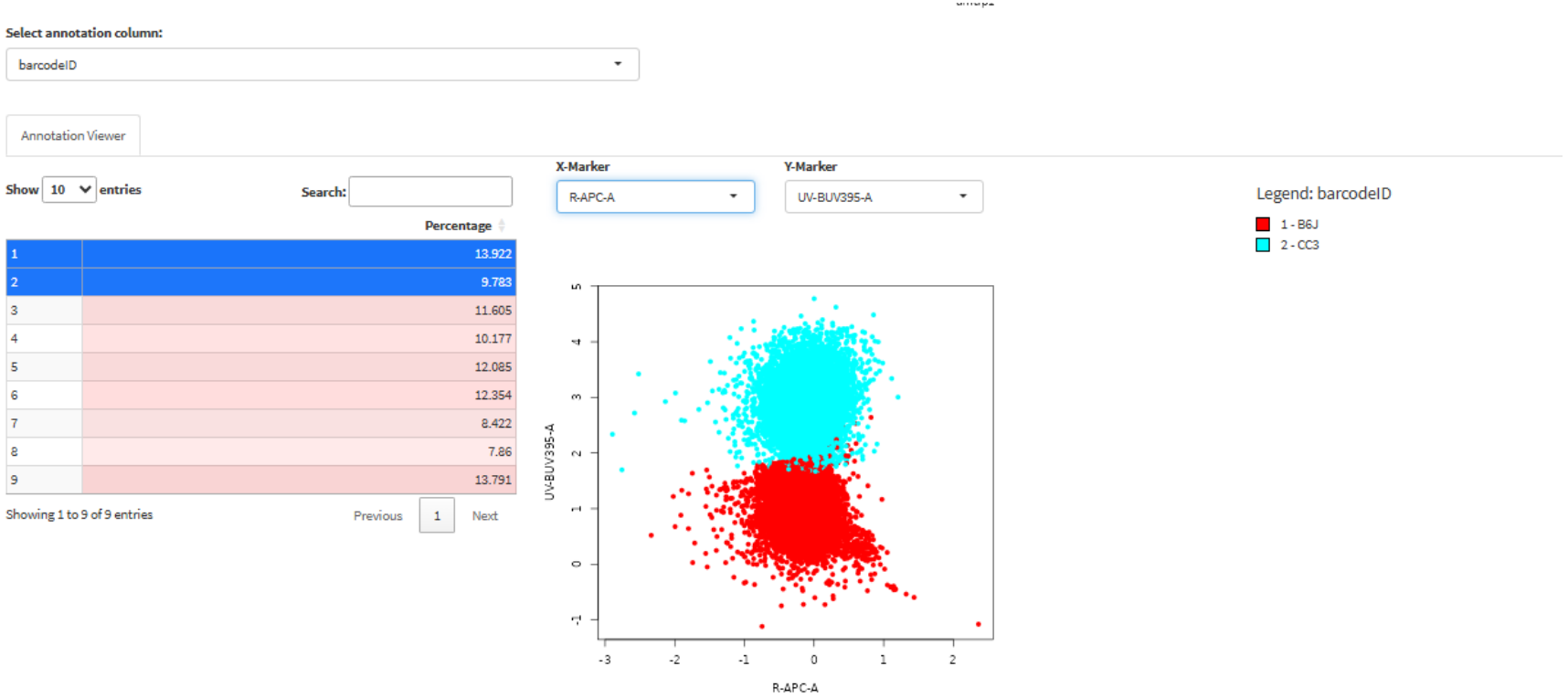


ANNOTATION QC

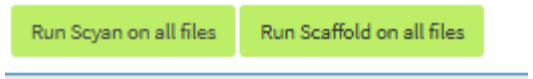
This section allows you to visually assess the quality of your annotations using UMAP projection.
A subset of all FCS files is used to generate a UMAP embedding.
Annotations (from Scyan or Scaffold) are overlaid on the UMAP for inspection.
If the clustering and annotation look correct, you can apply the annotation to all files by clicking the corresponding button.
This step helps verify that the model-based annotations align with known population structures before full application.



This section allows you to visually assess the quality of your annotations using UMAP projection.
A subset of all FCS files is used to generate a UMAP embedding.
Annotations (from Scyan or Scaffold) are overlaid on the UMAP for inspection.
If the clustering and annotation look correct, you can apply the annotation to all files by clicking the corresponding button.
This step helps verify that the model-based annotations align with known population structures before full application.



If you are satisfied with the results, click on 'Run on all files' to apply the annotation to all FCS files across your plates.



CIPHE CMP app:
pyInfinity Flow

Help – pyInfinityFlow

The **pyInfinityFlow** module allows you to predict unmeasured markers based on backbone markers using a deep learning model.

Step 1 – Data setup:

- For each plate, upload a TXT file containing plate information.
- This file must contain two tab-separated columns:
  - Infinity_target
  - Infinity_isotype

Step 2 – Marker selection:

- Select core panel markers as **backbone** markers (used for prediction).
- Designate the markers you want to impute as **exploratory**.

Step 3 – Run pyInfinityFlow:

- Select the number of cores to use, based on your machine's capabilities.
- Click **Run pyInfinityFlow** to start the prediction process.

Step 4 – Statistics:

- This part allows you to compute statistics either on:
  - The original measured values, or
  - The predicted values from pyInfinityFlow.
- First, define the positivity threshold for each ISO file to interpret marker expression

pyInfinityFlow

pyInfinityFLow :

STEP 1 : Please upload Infinity Markers

Upload for plate: 1 (TXT format)

Browse... No file selected

Upload for plate: 2 (TXT format)

Browse... No file selected

Upload for plate: 3 (TXT format)

Browse... No file selected

STEP 2 : Please select backbone and exploratory markers :

	name	desc	type
$P1	FSC-A	FSC-A	backbone
$P2	FSC-H	FSC-H	backbone
$P3	FSC-W	FSC-W	backbone
$P4	SSC-A	SSC-A	backbone
$P5	SSC-H	SSC-H	backbone
$P6	SSC-W	SSC-W	backbone
$P7	UV-BUV395-A	UV-BUV395-A	discard
$P8	UV-BUV496-A	UV-BUV496-A	backbone
$P9	UV-BUV737-A	UV-BUV737-A	backbone
$P10	UV-BUV805-A	UV-BUV805-A	backbone
$P11	V-BV421-A	V-BV421-A	backbone
$P12	V-V500-A	V-V500-A	backbone
$P13	V-BV610-A	V-BV610-A	backbone
$P14	V-BV650-A	V-BV650-A	discard
$P15	V-BV711-A	V-BV711-A	backbone
$P16	B-FITC-A	B-FITC-A	backbone
$P17	B-PE-Cy5-5-A	B-PE-Cy5-5-A	backbone
$P18	G-PE-A	G-PE-A	exploratory
$P19	G PE-TR-A	G PE-TR-A	backbone
$P20	G-PE-Cy5-A	G-PE-Cy5-A	backbone
$P21	G-PE-Cy7-A	G-PE-Cy7-A	backbone
$P22	R-APC-A	R-APC-A	discard
$P23	R-Alexa700-A	R-Alexa700-A	backbone
$P24	R APC-Cy7-A	R APC-Cy7-A	backbone
$P25	Time	Time	discard
$P26	vaevictis1	vaevictis1	discard
$P27	vaevictis2	vaevictis2	discard
$P28	barcodeId	barcodeId	discard
$P29	Flag	Flag	discard
$P30	fileId	fileId	discard
$P31	FlagApp	FlagApp	discard

STEP 3 : Run pyInfinityFlow

n cores

1 15 38

1 5 9 13 17 21 25 29 33 3738

Transform with pyInfinity flow method (est.logicle)

Run (py) InfinityFlow

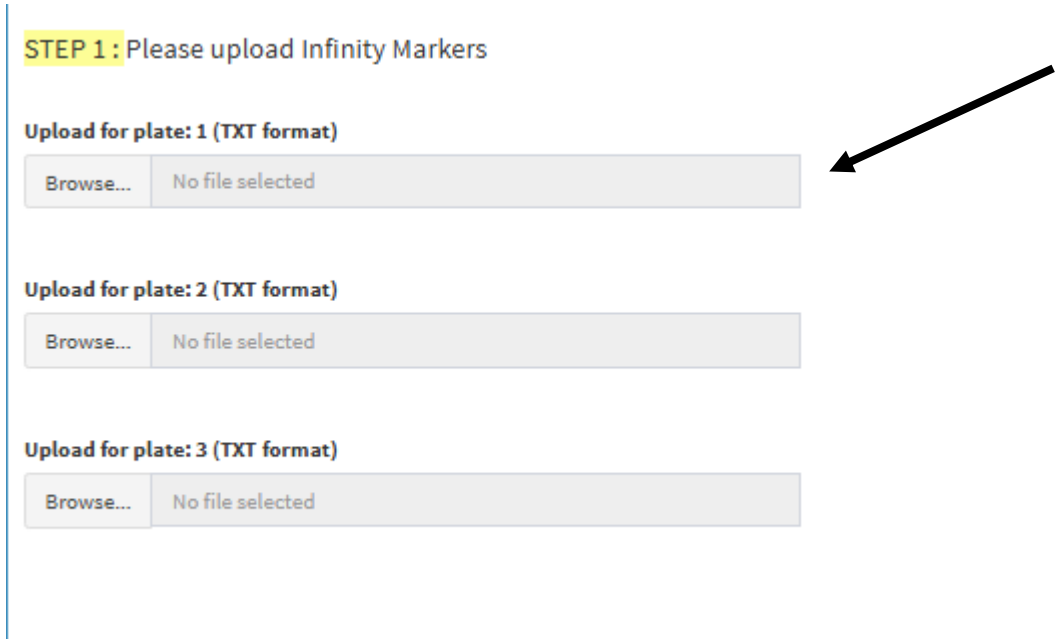
1-pyInfinityFlow

Step 1 – Data setup:

•For each plate, upload a TXT file containing plate information.

•This file must contain two tab-separated columns:

- •Infinity_target
- •Infinity_isotype




```
Infinity_target Infinity_isotype
"Rat IgG1,k"    "Rat IgG1,k"
 CD369 (Dectin-1/CLEC7A)        "Rat IgG1,k"
 PIR-A/B        "Rat IgG1,k"
 CD22   "Rat IgG1,k"
CD324 (E-Cadherin)      "Rat IgG1,k"
 CD172a (SIRPa) "Rat IgG1,k"
 CD319  "Rat IgG1,k"
"Rat IgG2a,k"   "Rat IgG2a,k"
 MAIR-V "Rat IgG2a,k"
 CD146  "Rat IgG2a,k"
PD-1H (VISTA)   "Rat IgG2a,k"
 CD8a   "Rat IgG2a,k"
" CD275 (B7-H2, B7-RP1, ICOS Ligand)"   "Rat IgG2a,k"
 Ly-6A/E (Sca-1)        "Rat IgG2a,k"
 CD40   "Rat IgG2a,k"
CD45R/B220      "Rat IgG2a,k"
 CD197 (CCR7)   "Rat IgG2a,k"
 CD47   "Rat IgG2a,k"
 CD98 (4F2)     "Rat IgG2a,k"
 CD14   "Rat IgG2a,k"
 CD107a (LAMP-1)        "Rat IgG2a,k"
 CD18   "Rat IgG2a,k"
 Ly-6G  "Rat IgG2a,k"
 CD21/CD35 (CR2/CR1)    "Rat IgG2a,k"
Mac-2 (Galectin-3)      "Rat IgG2a,k"
 CD199 (CCR9)   "Rat IgG2a,k"
 Ly-51  "Rat IgG2a,k"
 IgD    "Rat IgG2a,k"
 Tim-4  "Rat IgG2a,k"
 CD71   "Rat IgG2a,k"
```

1-pyInfinityFlow

STEP 2 : Please select backbone and exploratory markers :

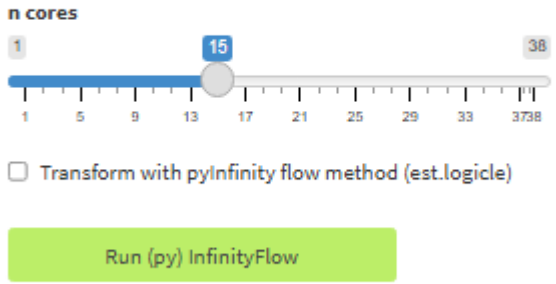
	name	desc	type
$P1	FSC-A	FSC-A	backbone
$P2	FSC-H	FSC-H	backbone
$P3	FSC-W	FSC-W	backbone
$P4	SSC-A	SSC-A	backbone
$P5	SSC-H	SSC-H	backbone
$P6	SSC-W	SSC-W	backbone
$P7	UV-BUV395-A	UV-BUV395-A	discard
$P8	UV-BUV496-A	UV-BUV496-A	backbone
$P9	UV-BUV737-A	UV-BUV737-A	backbone
$P10	UV-BUV805-A	UV-BUV805-A	backbone
$P11	V-BV421-A	V-BV421-A	backbone
$P12	V-V500-A	V-V500-A	backbone
$P13	V-BV610-A	V-BV610-A	backbone
$P14	V-BV650-A	V-BV650-A	discard
$P15	V-BV711-A	V-BV711-A	backbone
$P16	B-FITC-A	B-FITC-A	backbone
$P17	B-PE-Cy5-5-A	B-PE-Cy5-5-A	backbone
$P18	G-PE-A	G-PE-A	exploratory
$P19	G PE-TR-A	G PE-TR-A	backbone
$P20	G-PE-Cy5-A	G-PE-Cy5-A	backbone
$P21	G-PE-Cy7-A	G-PE-Cy7-A	backbone
$P22	R-APC-A	R-APC-A	discard
$P23	R-Alexa700-A	R-Alexa700-A	backbone
$P24	R APC-Cy7-A	R APC-Cy7-A	backbone
$P25	Time	Time	discard
$P26	vaevictis1	vaevictis1	discard
$P27	vaevictis2	vaevictis2	discard
$P28	barcodeId	barcodeId	discard
$P29	Flag	Flag	discard
$P30	fileId	fileId	discard
$P31	FlagApp	FlagApp	discard

Step 2 – Marker selection:

- Select core panel markers as **backbone** markers (used for prediction).
- Designate the markers you want to impute as **exploratory**.

1-pyInfinityFlow

Please indicate the number of cores and check box if you want to transform your data with estimate logicle. If you have already transform your data in section 2-Preprocessing, please do not check box



Step 3 – Run pyInfinityFlow:

- Select the number of cores to use, based on your machine's capabilities.
- Click **Run pyInfinityFlow** to start the prediction process.

2- Stats pyInfintiyFlow

Step 4 – Statistics:

•This part allows you to compute statistics either on:

- The original measured values, or
- The predicted values from pyInfinityFlow.

First, define the positivity threshold for each ISO file to interpret marker expression.

2- Stats pyInfintiyFlow

Calculate and export stats for PE mesured or estimated by pyInfinityFlow

Stats :

Missing values is when there are 0 positive cells (above ISO threshold) :

Select :

PE mesured

Show 10 entries Search:

Marker	threshold.pos	barcodeId	popID	count.pos	mean.pos
PIR-A/B	1000	12	8		
PIR-A/B	1000	16	8	1	2408.09
PIR-A/B	1000	3	8		
PIR-A/B	1000	60	8	1	1724.11
PIR-A/B	1000	71	8		
PIR-A/B	1000	8	8		
PIR-A/B	1000	80	8	1	1052.92
CD22	1000	6	8	5	2946.68
CD22	1000	10	8	7	1441.96
CD22	1000	12	8	16	1511.31

Showing 21 to 30 of 859 entries Previous 1 2 3 4 5 … 86 Next

Export stats

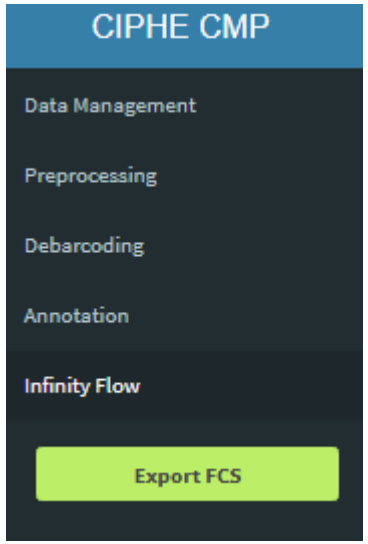
Marker	threshold.pc	barcodeId	popID	count.pos	mean.pos	median.pos	Perc.PE.pos	standard.dev	p5.pos	p25.pos	p75.pos	p95.pos
Rat IgG1,k	1000	6	8									
Rat IgG1,k	1000	10	8									
Rat IgG1,k	1000	12	8									
Rat IgG1,k	1000	16	8									
Rat IgG1,k	1000	3	8									
Rat IgG1,k	1000	60	8									
Rat IgG1,k	1000	71	8									
Rat IgG1,k	1000	8	8									
Rat IgG1,k	1000	80	8									
CD369 (Dect	1000	6	8									
CD369 (Dect	1000	10	8									
CD369 (Dect	1000	12	8	1	1403.23	1403.23	0.17		1403.23	1403.23	1403.23	1403.23
CD369 (Dect	1000	16	8	6	1432.96	1109.66	0.37	747.42	1019.83	1035.07	1345.05	2547.13
CD369 (Dect	1000	3	8									
CD369 (Dect	1000	60	8	1	1042.67	1042.67	1.05		1042.67	1042.67	1042.67	1042.67
CD369 (Dect	1000	71	8									
CD369 (Dect	1000	8	8									
CD369 (Dect	1000	80	8									
PIR-A/B	1000	6	8									
PIR-A/B	1000	10	8	1	1040.97	1040.97	0.35		1040.97	1040.97	1040.97	1040.97
PIR-A/B	1000	12	8									
PIR-A/B	1000	16	8	1	2408.09	2408.09	0.06		2408.09	2408.09	2408.09	2408.09
PIR-A/B	1000	3	8									
PIR-A/B	1000	60	8	1	1724.11	1724.11	0.97		1724.11	1724.11	1724.11	1724.11
PIR-A/B	1000	71	8									
PIR-A/B	1000	8	8									
PIR-A/B	1000	80	8	1	1052.92	1052.92	0.25		1052.92	1052.92	1052.92	1052.92
CD22	1000	6	8	5	2946.68	1621.85	0.64	2598.71	1267.62	1494.66	2972.04	6541.6
CD22	1000	10	8	7	1441.96	1279.13	3.4	342.15	1083.47	1214.35	1720.41	1888.55
CD22	1000	12	8	16	1511.31	1555.48	2.94	271.98	1156.55	1295.44	1718.41	1877.28
CD22	1000	16	8	44	1856.13	1719.18	3.05	646.07	1081.14	1441.28	2120.77	3125.29
CD22	1000	3	8									
CD22	1000	60	8	7	1710.64	1489.15	8.43	567.49	1189.23	1342.25	1932.27	2575.18
CD22	1000	71	8									
CD22	1000	8	8									
CD22	1000	80	8	25	2092.75	2033.98	7.16	744.68	1196.61	1562.55	2344.83	3424.34
CD324 (E-Cac	1000	6	8	3	2438.53	1309.52	0.33	2167.61	1092.57	1189	3123.56	4574.79
CD324 (E-Cac	1000	10	8	2	1503.85	1503.85	0.76	497.11	1187.49	1328.09	1679.61	1820.21

PE mesured PE estimated (XGBoost)

CIPHE CMP app: Backups

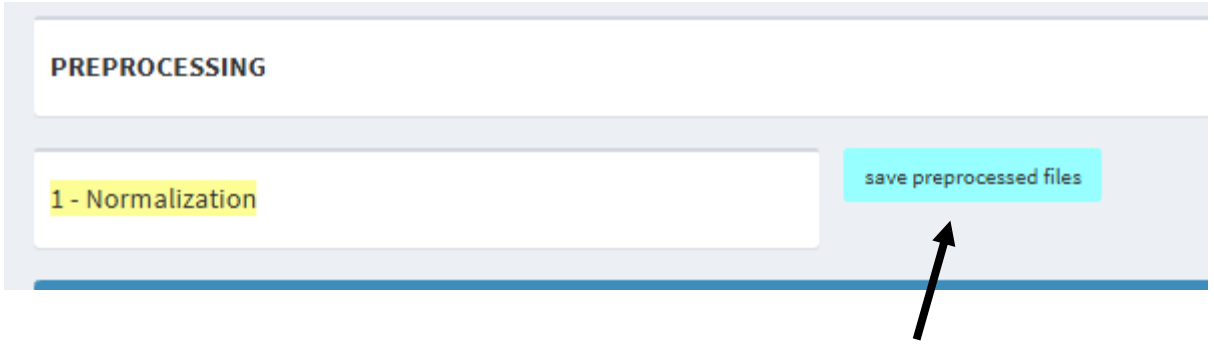
Backups

1 – Save FCS files



Useful for saving only the FCS files of one or more plates for further analysis in another tool.

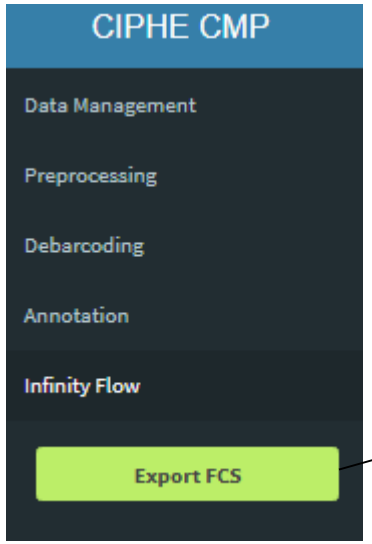
2 – Save project

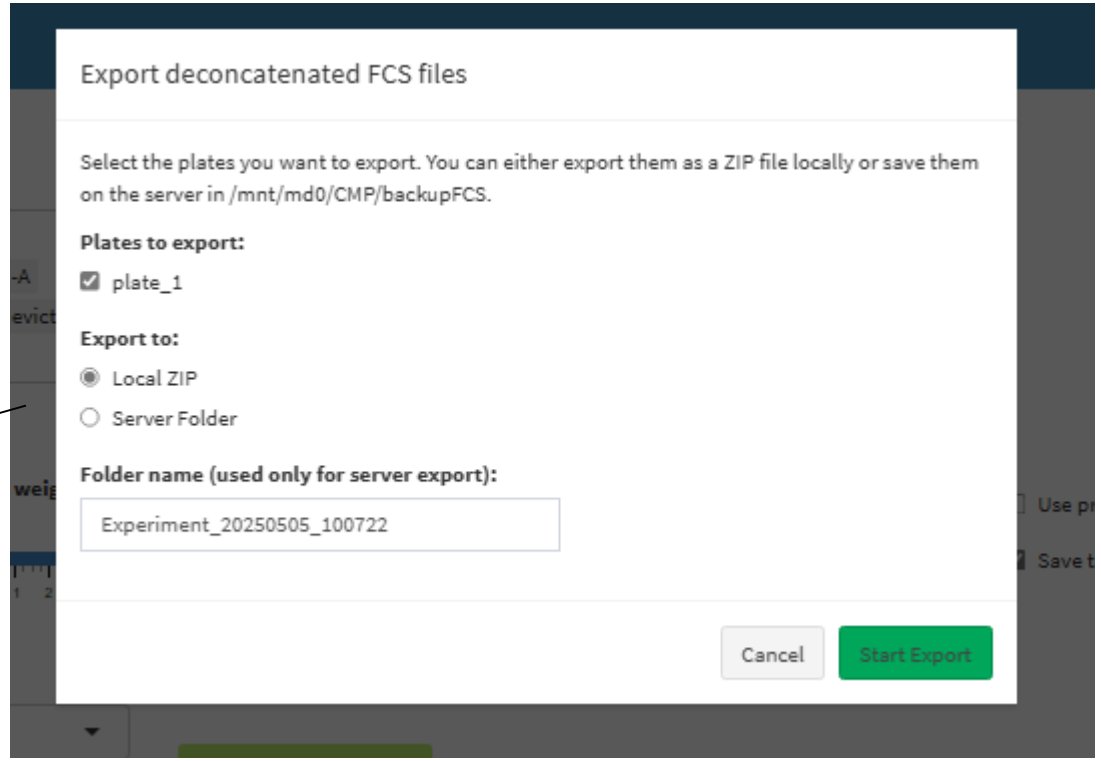


Saves the entire project and everything you've done in the application, so you can continue the analysis later.

Backups

1 – Save FCS files

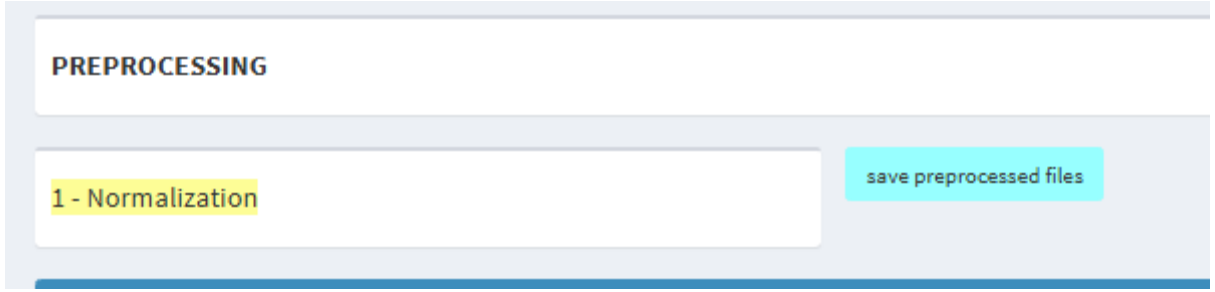




Useful for saving only the FCS files of one or more plates for further analysis in another tool.

Backups

2 – Save project



Saves the entire project and everything you've done in the application, so you can continue the analysis later.

You'll find save buttons for each section (I recommend saving after each minimum section).