BECKMAN COULTER
Life Sciences

An Automated FCS File Import and FlowSOM run in the Cytobank Platform Using the Application Programming Interface

In this Application Note you will learn:

- What an Application Programming Interface (API) is and what it is used for
- Which steps you need to take to start using the Cytobank API
- How to use the API of the Cytobank platform to upload your newly acquired FCS files automatically into a new Cytobank experiment and run a FlowSOM analysis from a previous template

Introduction

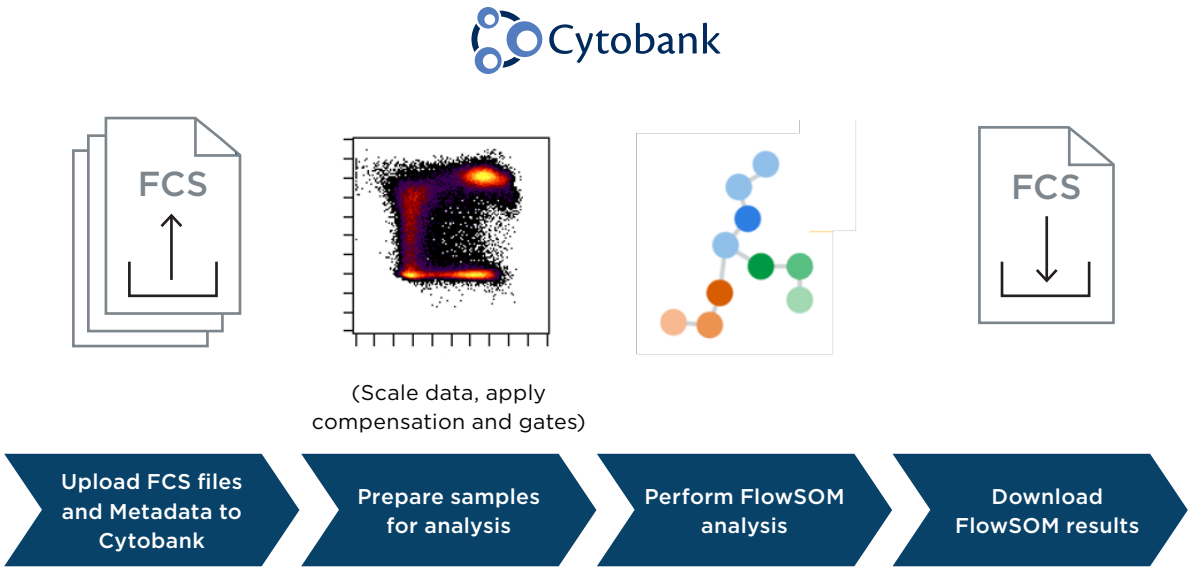
API stands for Application Programming Interface. An API is the set of instructions and available options for how two pieces of software can interact. Just as a graphical user interface (GUI) offers a visual interface for humans to interact with a software application, an API offers a non-graphical interface that other software applications can use to interact with a particular software application.

With the API, the Cytobank cloud can be accessed and used by any script or application. Regardless of the language used — R, Python, Matlab, Java, Ruby, Perl, etc. — data and configurations can be programmatically pulled from or pushed to Cytobank via the API.

There are many benefits to interface with the Cytobank platform using the API: since each experiment has its own design, through the API you can integrate the specific tools that address your needs. For instance, from the API you can extend the functionality of the Cytobank platform by integrating different tools for data analysis such as other dimensionality reduction algorithms or scripts for data clean up. As another example, if you are conducting a longitudinal study where you use the same panel for testing several patients over time, you can create a script that allows you to transfer the settings of scales, compensation and gating from a “template experiment” to all the subsequent ones. As a third example, suppose you manage many different projects where acquired data is saved to a server. You can connect the server to the Cytobank platform and use a script that detects when those files are generated by the instrument automatically upload them to Cytobank, saving you few manual steps that, if applied to several different projects, may take a lot of your time.

Other advantages of using the API could be to automatically share a long list of experiments with a new colleague or maybe set up parallel runs of the same algorithm with slightly different settings in order to find the best fit for your data set. Doing so manually can become tedious and error-prone, while coding will save you time and let you be sure about your results.

In this Application Note we will show you a script example that can automatically import FCS files from your local computer into the Cytobank platform, and using a templated experiment, transfer scales settings, gating, and run a FlowSOM.



Before doing this, we need to make sure you have all the tools you need to implement the script.

Step-by-step instructions to set up a programming environment for running the Cytobank API R package in RStudio

The following example is implemented using R. To use Cytobank API with other programming languages, please refer to the Cytobank API documentation online (**https://support.cytobank.org/hc/en-us/articles/115001546727-Cytobank-API-Libraries-for-Programming-Languages**).

Before being able to use packages in R to interface with Cytobank, such as the CytobankAPI and CytobankBridgeR packages, there are several prerequisites that will enable you to set up a working R environment to start using such packages:

1. The latest version of R & RStudio

Download and install a working version of R from the official R website and choose the correct operating system.

Recommended: Download RStudio, an integrated development environment (IDE) that makes it easy to use R once a working version of R is installed.

Use these links:

https://www.r-project.org/

https://rstudio.com/products/rstudio/download/#download

Note that this must only be done once.

2. Install essential Cytobank packages

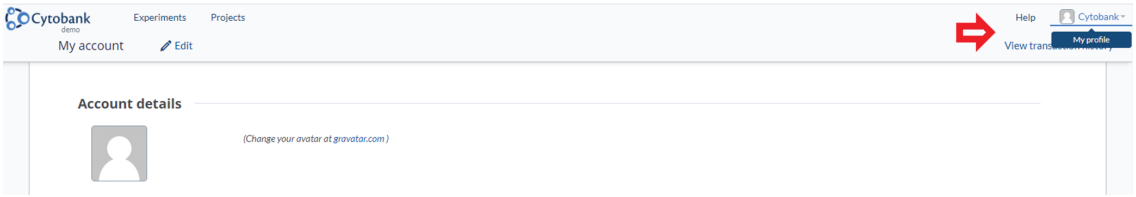
Once a working version of R is installed, open up RStudio (or R) and install the CytobankAPI package via the following console command:

utils::install.packages("CytobankAPI")

! please note that the package name is case-sensitive and the quotation marks for the package name are required

3. Get your API token

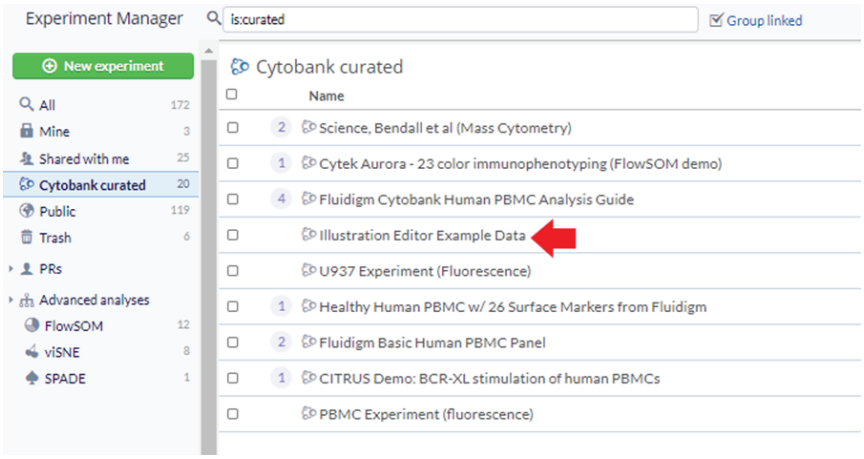
Go to your Cytobank user profile:



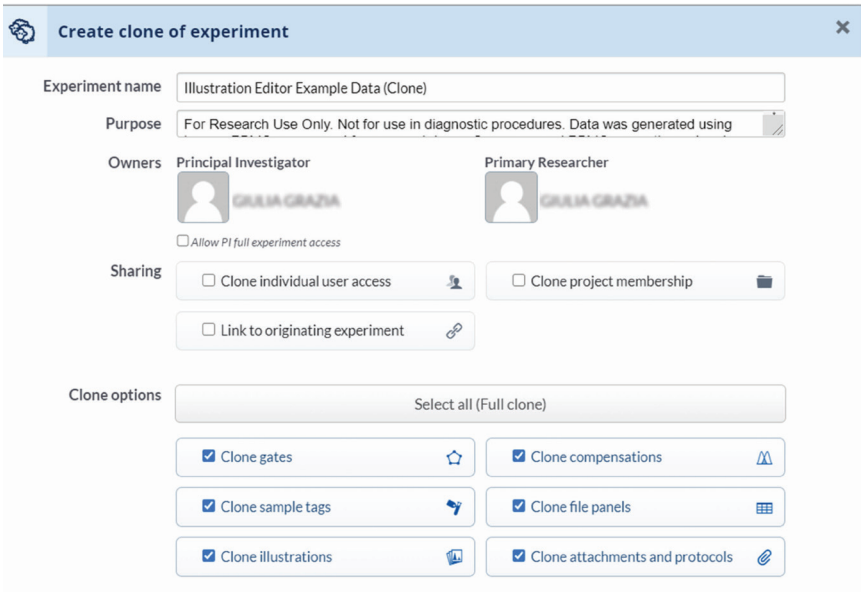
Scroll down the page to generate API token (it expires after 8 hours).

Download FCS files and gating ML we will need for this exercise

Access your Cytobank platform and search among the "Cytobank curated" experiments the one called **Illustration Editor Example Data**



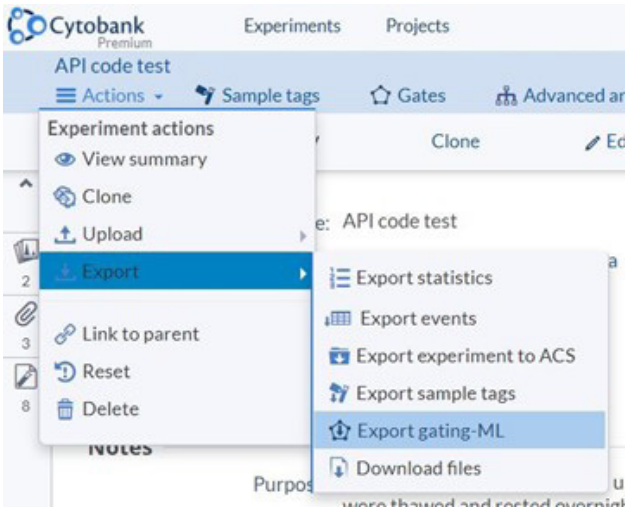
Clone your own copy (please make sure to check the "gates" but uncheck the project membership as well as individual user access)



Download the FCS files (please remember to unzip if needed)

Note the experiment ID of your cloned experiment. To learn more read **support article**.

Download the gating-ML file from this experiment (Actions > Export > Export gating-ML). To learn more read **support article**.



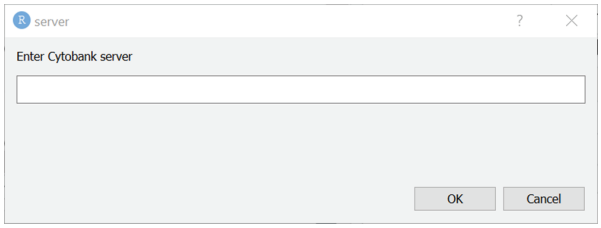
Here we provide an example of a script to automatically upload FCS files from a user-defined folder to a new Cytobank experiment using RStudio.

```
# Install and load packages ------------------------------------

# If you do not have packages tidyverse and rstudioapi installed, install them as
follows:

install.packages('rstudioapi')

install.packages('tidyverse')

# Load packages------------------------------------------------------

library('CytobankAPI')

library('rstudioapi')

library('tidyverse')

# Set up credentials and experiment information --------------------

server = rstudioapi::showPrompt('server', 'Enter Cytobank server')

experiment_name = rstudioapi::showPrompt('experiment name', 'Experiment name')

purpose = rstudioapi::showPrompt('purpose', 'Experiment purpose')

auth_token = rstudioapi::showPrompt('token', 'API authentication')

# Authenticate -----------------------------------------------------

cyto_session = authenticate(server, auth_token = auth_token,

                            short_timeout = 30, long_timeout = 3000)

# Create experiment ------------------------------------------------

new_experiment <- experiments.new(cyto_session, experiment_name, purpose)

experiment_id <- new_experiment$id

# Select FCS file directory ----------------------------------------

fcs_files <- list.files(rstudioapi::selectDirectory(caption = 'Select directory containing
FCS files'),

                        pattern = '*.fcs$', full.names = T)

# Upload FCS files -------------------------------------------------

purrr::imap(fcs_files,

            ~{fcs_files.upload(cyto_session, experiment_id, fcs_files[.y]);

              cat(.y, '/', length(fcs_files), ' uploaded\n')

            })

# Open experiment --------------------------------------------------

browseURL(paste0('http://', server, '.cytobank.org/cytobank/experiments/', experiment_id))
```

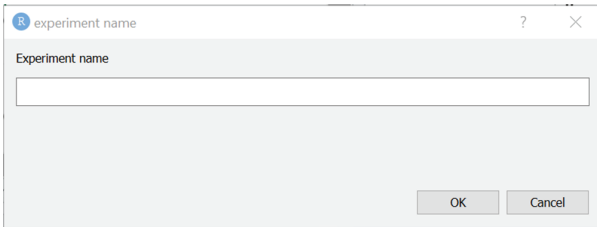
Note: It may take a moment to process the files after upload is complete. If you navigate to the experiment and you do not see the FCS files, wait a minute and refresh the page.

To run this script, copy it into R studio and then execute it line by line. You will then be prompted:

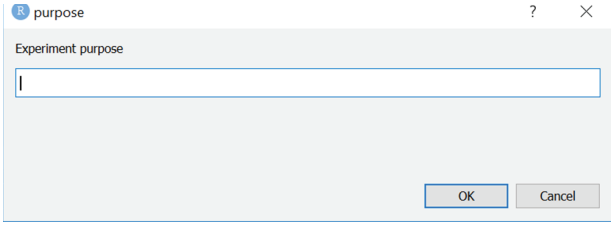
- To enter your Cytobank server (e.g., “premium”, “yourinstitutionname” if using an Enterprise server)



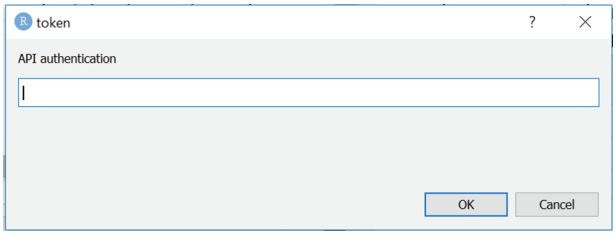
- Give the Experiment a name



- Give the Experiment a purpose



- Enter the API token



At the end, you will be asked to select the directory containing the FCS files you want to upload.

In the next section, you can follow along to learn how to set up an automated FlowSOM analysis. For the purpose of this App Note we will use:

- Standard settings
- Scales and gating from a template experiment (that must contain the same panel of channels

```
# Transfer scales from template experiment -----------------------

# Download scales from template experiment ========================

template_experiment_id <- rstudioapi::showPrompt('template', 'Enter experiment ID of
template experiment')

scales_template <- scales.list(cyto_session, template_experiment_id)

scales_default <- scales.list(cyto_session, experiment_id)

scales_updated <- bind_cols(scales_default['id'], select(scales_template, -id)) %>%

  mutate(experimentId=scales_default$experimentId)

# Upload scales to new experiment ================================

map(scales_updated$id, function(x) {

  scales.update(cyto_session, scale=scales_updated[x==scales_updated$id,])

  }

)

# Upload gates from gating-ML file -------------------------------

gatingML_file <- rstudioapi::selectFile('Select gating-ML to upload', 'Select')

# View gates in experiment. --------------------------------------

#  Check, make any adjustments as needed, and click "Apply" button.

browseURL(paste0('http://', server, '.cytobank.org/cytobank/experiments/', experiment_id,
'/gating'))

# Create FlowSOM analysis ----------------------------------------

flowsom_name <- rstudioapi::showPrompt('FlowSOM name', 'Enter FlowSOM name')

new_flowsom <- flowsom.new(cyto_session, experiment_id, flowsom_name)
```

```
# Choose FlowSOM settings ----------------------------------------

new_flowsom@selected_population_name <- 'lymphocytes'

new_flowsom@fcs_files <- fcs_files.list(cyto_session, experiment_id)$id

clustering_markers <- c('CD3', 'IFNg', 'CD137', 'HLA-DR', 'PD-1', 'CCR7', 'TNFa', 'CD4',
'CD8', 'CD16', 'CD14', 'CD45RA')

new_flowsom@channels <-new_flowsom@.available_channels$`Panel 1`$channels %>%

  data.frame() %>%

  dplyr::filter(str_detect(longName,paste0(clustering_markers, collapse = '|'))) %>%

  pull(normalizedShortNameId)

new_flowsom@population_id <- unlist(new_flowsom@.available_populations[new_flowsom@.
available_populations$name==new_flowsom@selected_population_name,]$gateSetId)

# Event sampling

new_flowsom@event_sampling_method <- 'equal'

new_flowsom@desired_events_per_file <- 25000

# Algorithm settings

new_flowsom@clustering_method <-'consensus'

new_flowsom@expected_metaclusters <- 12

new_flowsom@expected_clusters <- 100

new_flowsom@iterations <- 10

new_flowsom@random_seed <- sample(1:10^7, 1)

# Transformations

new_flowsom@normalize_scales <- F

# Update FlowSOM object with changes above

flowsom.update(cyto_session, new_flowsom)

# Run FlowSOM

flowsom.run(cyto_session, new_flowsom)

# Check FlowSOM status

flowsom.status(cyto_session, new_flowsom)

# When FlowSOM status is complete, open FlowSOM result experiment

created_experiment <- flowsom.status(cyto_session, new_flowsom)$createdExperiment

browseURL(paste0('https://', server, '.cytobank.org/cytobank/experiments/', created_
experiment))
```

Once done uploading FCS, your browser will open and you will see your new Cytobank experiment (with the name you chose and containing the FCS files from the folder you pointed at) and the page where you can download the results from the FlowSOM run.

TIPS and TRICKS:

- If you want to identify all the programming languages available to you, visit **https://developer.cytobank.org/**
- If you want to find out more about the R package CytobankAPI, it is available on CRAN with reference, quickstart, advanced analysis vignette **https://cran.r-project.org/web/packages/CytobankAPI/index.html**
- This tech note uses R Studio 1.4.1103
- Depending on the location of the server used, restrictions may apply and the described process may not be supported. For more details contact Cytobank Support.

For Beckman Coulter's worldwide office locations and phone numbers, please visit Contact Us at **beckman.com**

20.11.335.FLOW