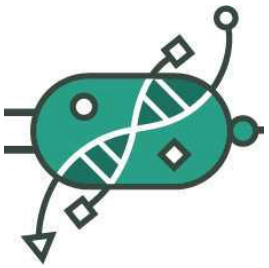
Jul 24, 2024 Version 3

OT-2 Protocol to transfer volume from several plates to a single plate V.3



DOI
dx.doi.org/10.17504/protocols.io.6qpvr4o62gmk/v3

Ana Mariya Anhel[1], Lorea Alejaldre[1], Ángel Goñi-Moreno[2]

[1]Centro de Biotecnología y Genómica de Plantas, Universidad Politécnica de Madrid (UPM)-Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria (INIA/CSIC), Madrid, Spain;
[2]Centro Nacional de Biotecnología (CNB-CSIC)

Ángel Goñi-Moreno: angel.goni@cnb.csic.es;

biocomp.cbgp Biocomputation Lab
Centro de Biotecnologia y Genomica de Plantas

DOI: dx.doi.org/10.17504/protocols.io.6qpvr4o62gmk/v3

Protocol Citation: Ana Mariya Anhel, Lorea Alejaldre, Ángel Goñi-Moreno 2024. OT-2 Protocol to transfer volume from several plates to a single plate . **protocols.io https://dx.doi.org/10.17504/protocols.io.6qpvr4o62gmk/v3**Version created by **Ana Mariya Anhel**

Protocol status: Working
We use this protocol and it's working

Created: November 20, 2023

Last Modified: July 24, 2024

Protocol Integer ID: 101406

Keywords: OT-2, Opentrons, Media transfer, 96-well

Funders Acknowledgement:
Comunidad de Madrid
Grant ID: Y2020/TCS-6555, 2019-T1/BIO-14053
MCIN/AEI
Grant ID: CEX2020-000999-S, PID2020-117205GA-I00
European Research Council
Grant ID: 101044360

Abstract

This protocol is meant to transfer samples from different plates to a single or fewer final plates than the source number, in other words, merge samples of different source plates in a final plate.
The output of running this script will be the final plate(s) with samples and a reactive (optional), and the corresponding map(s) with the original identities of the samples which will be given by the user in the input file.

This protocol uses a python script for an Opentrons 2 robot and an excel file containing the required variables to set the number of samples, volumes of transfer, type of plates, etc...

In our laboratory, this protocol has been used as part of the "High-throughput workflow for the genotypic characterization of transposon library variants" also available in protocols.io

This protocol is a set of instructions or description of the **LAP repository** entry **LAP-NplateMerging-OT2-2.0.0**
You can find the script and complementary information for this specific version of the protocol in this **LAP entry link** and **GitHub Link to LAP entry documents**

The major changes from previous version are:

- **Change of name variable API *Name Rack 15mL Falcon Reactives*** which is called now *API Name Rack Falcon Reactives*
- **The program now accepts falcon tubes and tube racks of 15ml or 50ml.** It does not accept mix tube racks.
- **Description of robot and protocol setup in a separate protocols.io entry** (Setting and Customizing OT-2 for LAP Entries)
- The page(s) that corresponds to the map(s) set in the variable **'Name Sheet Map Identifiers'** of the page PerPlateVariables **need to have the column and row names** corresponding to the ones of the labware set in the variable 'API Name Source Plate' of the sheet GeneralVariables

Guidelines

This protocol was developed with python 3.7.1, OT App Software Version 7.0.2 and API level version 2.14 in a Linux 4.14.74 system (these are the OT-2 specifications).
In the script several packages are used: pandas (0.25.3), openpyxl (3.1.2), math, random and numpy (1.15.1)

It has been tested with cultures from *Pseudomonas putida* KT2440 as part of the High-throughput workflow for the genotypic characterization of transposon library variants.

The maximum number of 96-well plates per run is 8 source plates to 1 final plates dispensing some media (need of 1 falcon tube rack), 1 pipette is required for all the volumes and the replace of tip racks is allowed (set as True in the input variable file)

Materials

Software

- Python 3.7.1
- opentrons software version 7.0.2
- python packages: pandas (0.25.3), openpyxl (3.1.2), numpy(1.15.1), math, random
- OT App
- Excel

OT-2 Labware

- Opentrons Tip racks

Equipment

Opentrons 96 Tip Rack 300 µL	NAME
Tip rack	TYPE
Opentrons	BRAND
-	SKU
https://labware.opentrons.com/opentrons_96_tiprack_300ul?category=tipRack	LINK

Equipment

Opentrons 96 Tip Rack 20 µL	NAME
Tip rack	TYPE
Opentrons	BRAND
-	SKU
https://labware.opentrons.com/opentrons_96_tiprack_20ul?category=tipRack	LINK

- 96- well plates

Equipment

96-well plates, flat bottom, non treated	NAME
Cell culture plates	TYPE
VWR	BRAND
734-2781	SKU
https://es.vwr.com/store/catalog/product.jsp?catalog_number=734-2781	LINK

- Opentrons Falcon tube rack

Equipment

Opentrons 15 Tube Rack with Falcon 15 mL Conical	NAME
OT Tube Rack	TYPE
Opentrons	BRAND
-	SKU
https://labware.opentrons.com/opentrons_15_tuberack_falcon_15ml_conical/	LINK

- 15mL Falcon tubes

Equipment	
Falcon® Conical Centrifuge Tubes 15mL	NAME
Flaocn Tube	TYPE
Falcon	BRAND
352096	SKU
https://ecatalog.corning.com/life-sciences/b2c/US/en/Liquid-Handling/Tubes,-Liquid-Handling/Centrifuge-Tubes/Falcon%C2%AE-Conical-Centrifuge-Tubes/p/falconConicalTubes	LINK

Equipment:

Equipment	
OT-2	NAME
Liquid handler	TYPE
Opentrons	BRAND
OT-2	SKU

Equipment

Single Channel Electronic Pipette (GEN2) 300uL	NAME
Opentrons Pipette	TYPE
Opentrons	BRAND
-	SKU
https://shop.opentrons.com/single-channel-electronic-pipette-p20/	LINK

Equipment

Single Channel Electronic Pipette (GEN2) 20uL	NAME
Opentrons Pipette	TYPE
Opentrons	BRAND
-	SKU
https://shop.opentrons.com/single-channel-electronic-pipette-p20/	LINK

Before start

It is important to note the location of the source plate in the OT-2 slots, the identity of the samples is provided by excel sheets in the input variable. It should be noted that If 2 samples within a plate have the same name, they will be indistinguishable in the final plate(s)

Files Preparation

1 **Preparing Customized Template**

Preparing the template (a .xlsx) with the specific variables for each experiment.

Attached there is a template of the variable file with several sheets and a PDF file explaining each variable:

1. **GeneralVariables:** variables related to the labware to be used
2. **PipetteVariables:** variables related to the pipettes to be used
3. **PerPlateVariables:** variables related to the specifications of each source plate
4. **Maps (Optional):** sheet(s) with the names of the samples in the source plates. These will be reflected in the final plate map *--> not included in the template but needs to be included and have the same names as established in the variable* ***Name Sheet Map Identifiers*** *from the PerPlateVariablesSheet*

MergeSamplesInstructionsv200.pdf

TemplateMergeSamples.xlsx

> Note
>
> The most updated Excel template can be found in the **LAPrepo Repository Page**

1.1 *Fill the template with the corresponding values*

1.2 *Save it with the name VariablesMergeSamples.xlsx*

> Note
>
> The file should be spelt **exactly** *VariablesMergeSamples.xlsx* or the Python script won't work correctly

Setting the robot

2 **Prepare the system of the robot to run the protocol**

For this protocol to work we need to transfer the *VariablesMergeSamples.xlsx* to the directory */data/user_storage* of the OT system that we will use to perform the protocol

As well, if we are using custom labware we need to upload it to the OT App and send it to the directory */data/labware/v2/custom_definitions/custom_beta* if the labware is not there yet.

Finally, we need to make sure the package *openpyxl* is installed in the robot system

We can do this entire step by following the protocol *Setting and Customizing OT-2 for LAP Entries* with the specifications given in the text above

Protocol

NAME

Setting and Customizing OT-2 for LAP Entries

CREATED BY

biocomp.cbgp Biocomputation Lab

PREVIEW

Running Protocol

3 **Load script in OT-App**

Now that we have transferred the variable files to the robot, we can import the script and run it in the selected robot

Note

This whole step has been developed and tested with version 7.0.2 of the OT-App

Indications may vary from version to version of the opentrons App and the version of the script.

Software

NAME	OS	DEVELOPER	SOURCE LINK
Opentrons App	Windows >=10, Mac >=10 , Ubuntu >=12.04	Opentrons	https://opentrons.com/ot-app/

3.1 *Load the script in the App*

Protocols -> Import -> Drag Python script

This version of the protocol was developed when the last version available of **LAP-NplateMerging-OT2** was the 2.0.0 which script you can find attached

ScriptMergePlates_v200.py

The name of the python file is user's choice, it will work with any name in the app.

Note

The last script version can be found at **https://github.com/BiocomputationLab/LAPrepository/tree/main/LAPEntries**. The name of the directory should be **LAP-NplateMerging-OT2** followed by the version.

As well we can find the latest version of the script at **https://www.laprepo.com/repository/** with the same name as in GitHub

Software

NAME	DEVELOPER	SOURCE LINK
LAP Repository	https://biocomputationlab.com/	www.laprepo.com

Note

The App with version 7.0.2 analyzes your protocol before setting a robot to run, so the labware will not be shown before assigning the protocol to a specific robot when you import it into the App

3.2 *Select Robot to Perform Script*

Click in the protocol -> Start setup -> Choose the OT where the file *VariablesMergeSamples.xlsx* is -> Proceed To Setup

After clicking on Proceed to Setup, you should obtain, the positions of the labware in the *Labware* and in the *Liquids* tabs, the reagents and their corresponding volume.

In case the protocol with the set variables cannot run, an error will occur during the run of that simulation. Many errors are contemplated already and have a specific message that hints the user what could have gone wrong.

Note

The volume of the initial samples is established to be 90% of the max volume of the well, but it is only a recommendation. **Just make sure that there is enough volume to transfer to the final plate(s).**

The volume of the reagents is exactly what is needed, so it is **suggested to pour always more to take in account the error of pipetting**

Note

It is recommended that you perform a labware position check.

You can do it with test plates after loading the script but before cleaning the surface. That way, you reduce the probability of contamination (using the test plates and labware) and pipetting errors (position check).

4 Run Protocol in OT

4.1 *Make sure the needed calibrations are done*

Pipettes, tip racks and tip length calibrations need to be done for the items used in this run

4.2 *Labware position check is performed (if required)*

4.3 *Clean* the *surface of* the *robot with 70% ethanol to clean and* disinfect *the surfaces*

Note

Check the Opentrons page https://support.opentrons.com/s/article/Cleaning-your-OT-2? for more information about cleaning the OT-2 robot with the proper materials

4.4 *Set the labware and reagents as shown in the OT-App*

4.5 *Start Run*

The procedure that the robot is going to do is mainly divided into 3 parts:

1. Select which samples, taking into account the variable *Type of Sample Selection*
2. (Optional) Distribute reactive(s)
3. Distribute selected samples

Expected result

One or more plates, with a reactive mixed with the samples from different sources or only the samples. In addition to an **Excel file where every sheet is the map of a final plate located in the folder *`/data/user_storage`*** that will give the position in these plates with their identifiers (the name given in the maps of each source plate in the input variable file)

After-Running

5 **Retrieve labware from the OT**

6 **Import map from robot**

There will be a map with the name set in the variable *Name File Final Map* in the sheet *GeneralVariables* followed by the extension .xlsx: [*Name File Final Map*]*.xlsx*

This file will be located in the directory */data/user_storage* in the robot where the script has been run.

To retrieve the file, we can go to step #2 Linked protocol and reproduce it by transferring the files from the robot to the computer.

Take in account that files overwrite, so if you have given the same name to the final map in 2 consecutive runs, you will get only the results of the last run.

Expected result

The sheets in the final map contain the identity of the samples selected in the places where they have been placed/distributed

These identities are going to be the names given in the map Excel sheets from the input file

Example

1h 5m

7 **We want to merge 16 samples from 5 source plates in a different way, 1 of them selecting samples from back to front, 2 in a random way and 2 from front to back.**
Our final plate will have half the plate already occupied by other samples, and we want to add to all the samples water, so we need to take into account those details

We will use a computer with a Windows 10 system

7.1 **Prepare variable file** 10m

Excel template that we can find go to step #1 filled and saved with the name *VariablesMergeSamples.xlsx*

VariablesMergeSamples.xlsx

7.2 **Upload custom labware to app** 2m

We are using a custom labware called *vwrblueprintdepth105_96_wellplate_390ul* that has been created with the labware creator that opentrons offers (**https://labware.opentrons.com/create/**)

vwrblueprintdepth105_96_wellplate_… 11KB

We upload it to the opentrons app (make sure that is in the robot app) and the robot system as stated in the protocol in step Setting and Customizing OT-2 for LAP Entries

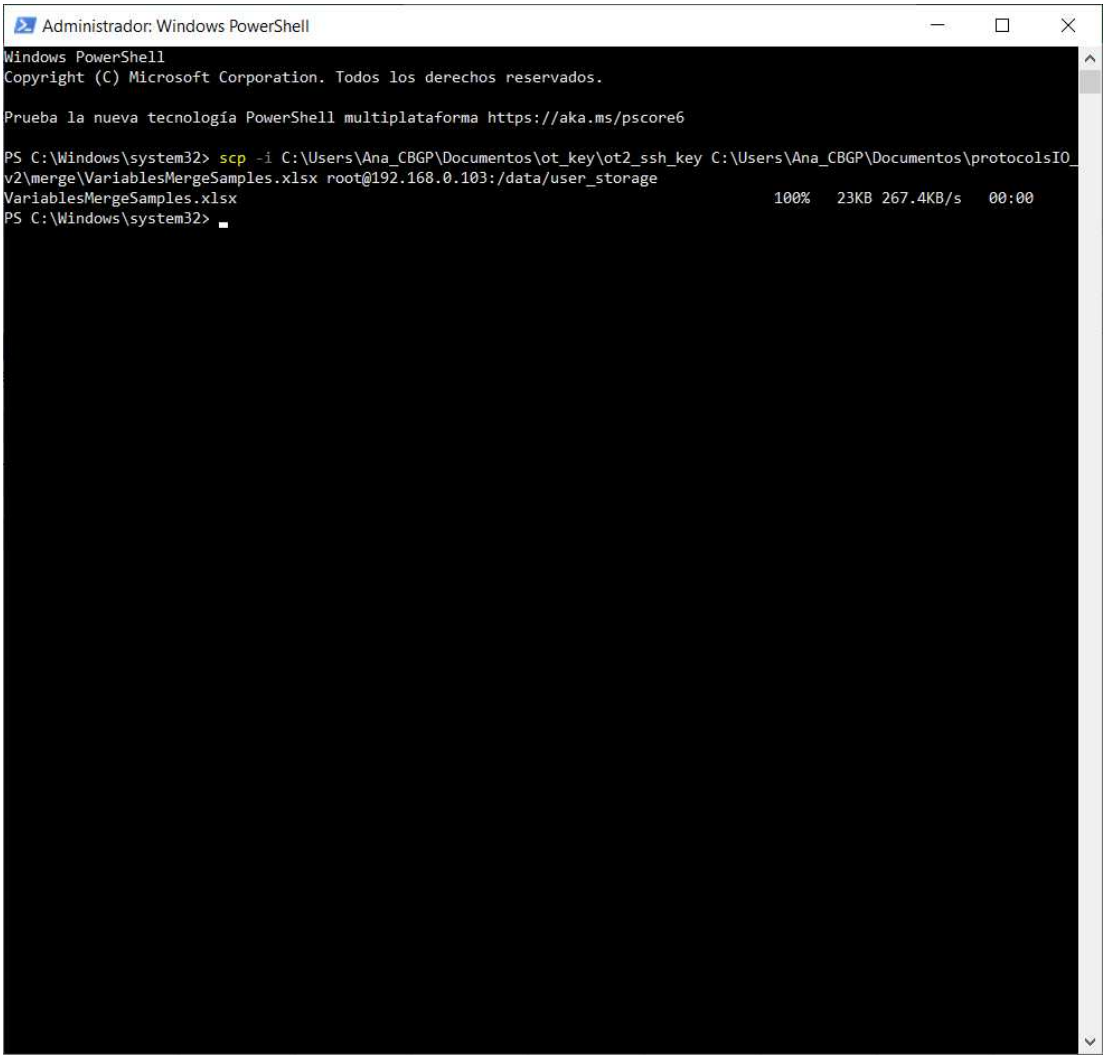
vwrblueprintdepth105_96_wellplate_…

(this is a zip file because needed to be compressed to upload to protocols.io but what needs to be transferred to the robot is the folder inside of the zip file)

7.3 **Export the variable file to *the /data/user_storage* folder in the robot** 2m

For more information about sending files to the OT-2

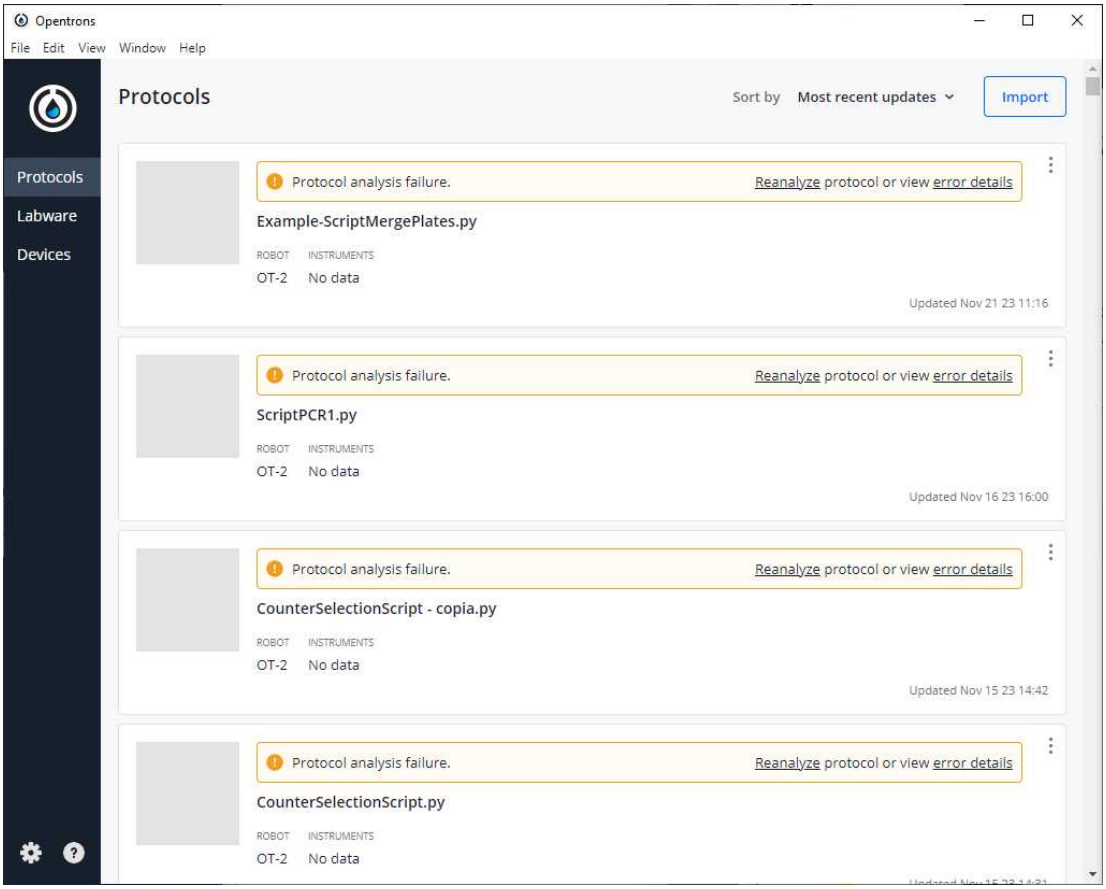
Setting and Customizing OT-2 for LAP Entries



command line window with scp commands to transfer the variables .xlsx from our computer to the OT-2

7.4 **Import the script** that we have downloaded from the step go to step #3.1 (I named it *Example-ScriptMergePlates.py*) to the OT-App 30s

Example-ScriptMergePlates.py



Result of importing the Python script into the OT-App

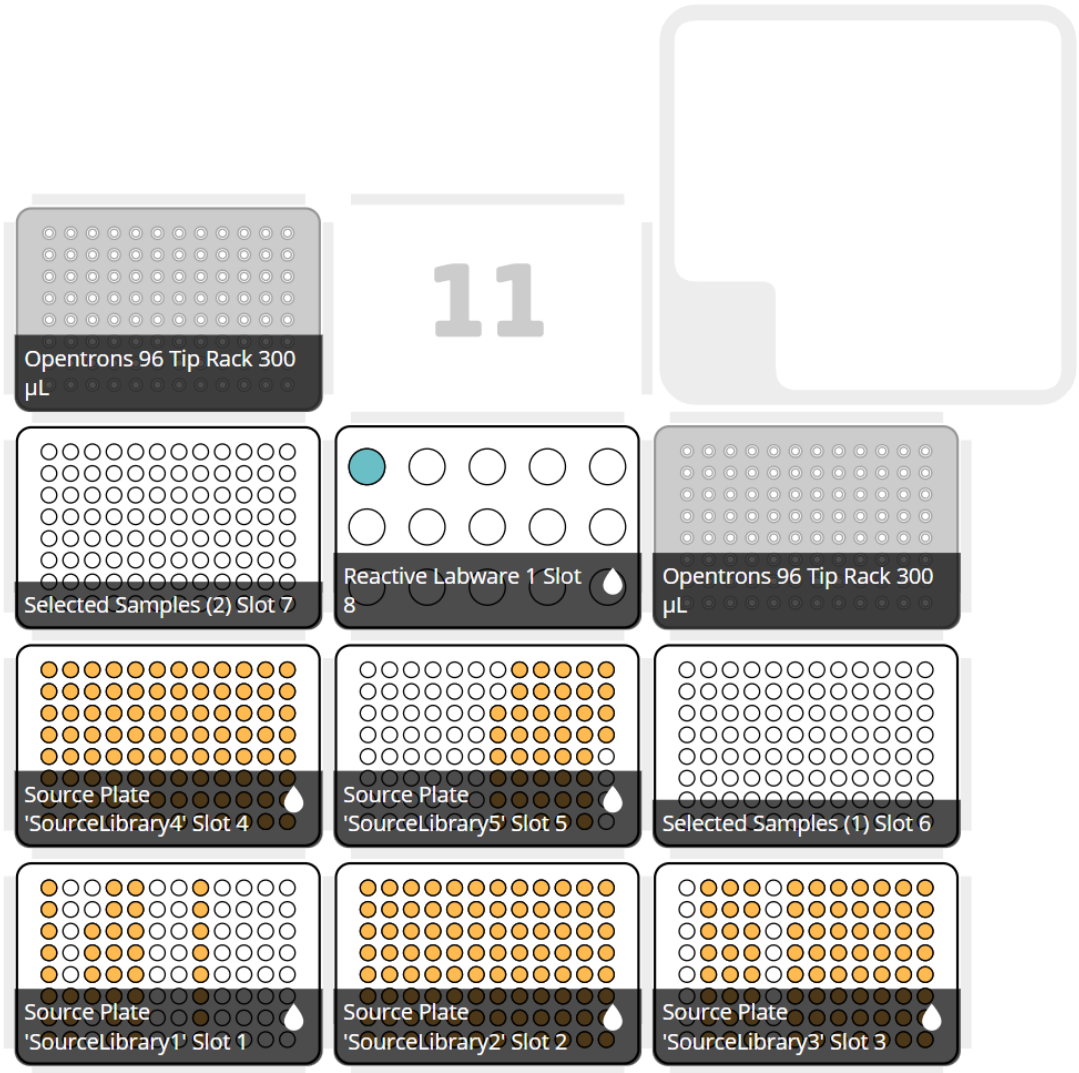
As we can see, we have an error, but that is programmed because the script is meant to work in the robot but not in your computer

7.5 Run the protocol in the robot that we have transferred the Excel file

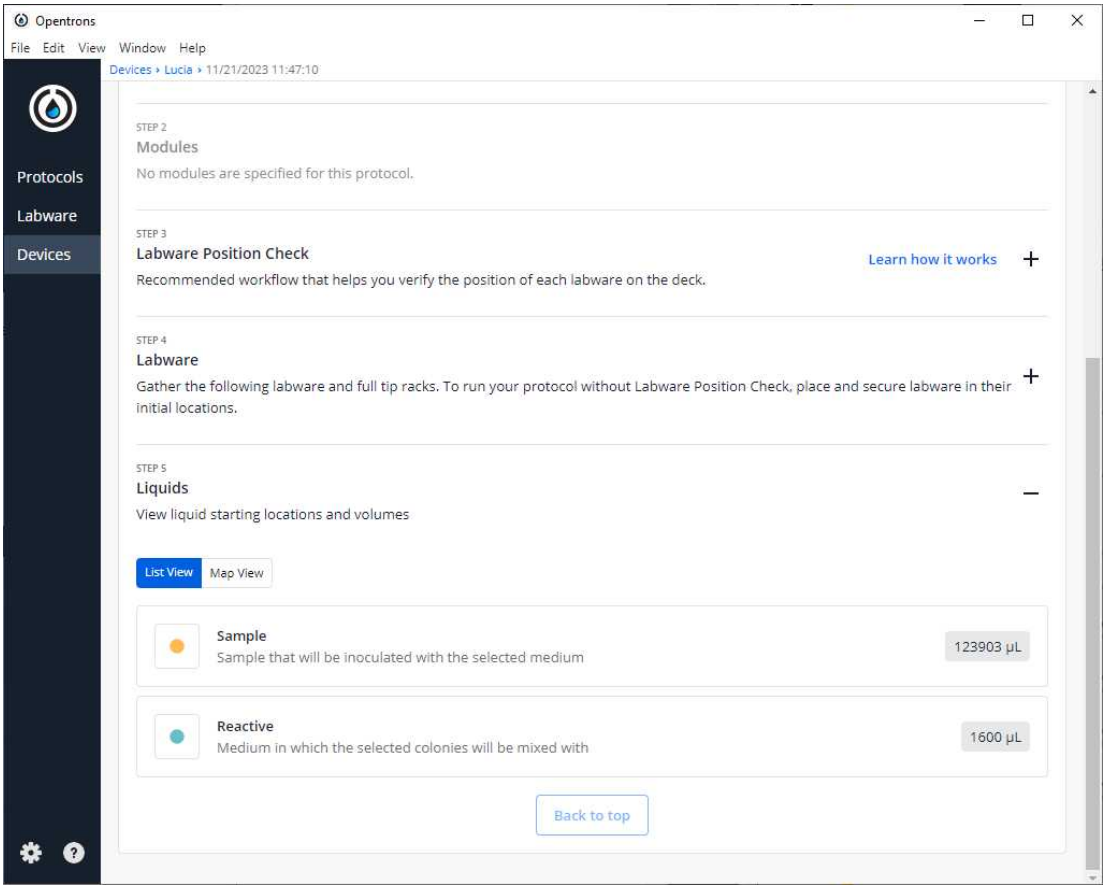
1m

***Example-ScriptMergePlates.py* -> Start setup -> Select the robot** in which we are going to run the protocol

If we do not have any errors, the output should look similar to the following pictures



Labware and liquid set-up layout



Volumes of the needed liquids to perform the protocol

7.6 **Turn the HEPA filter module** 30s

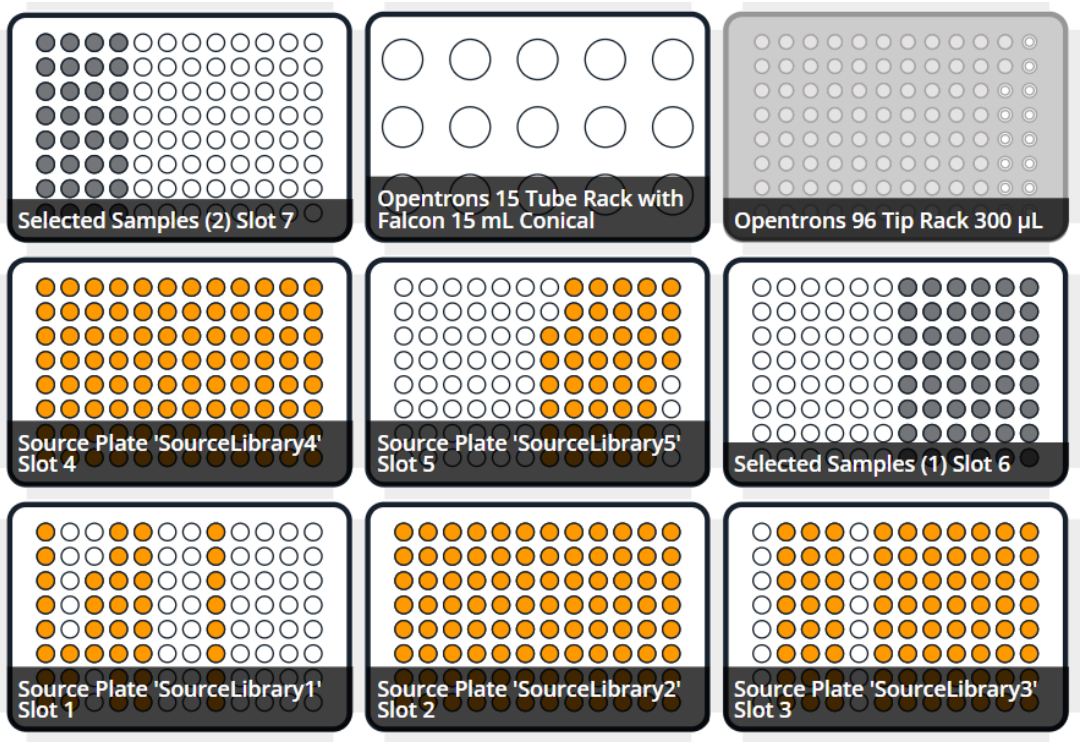
*

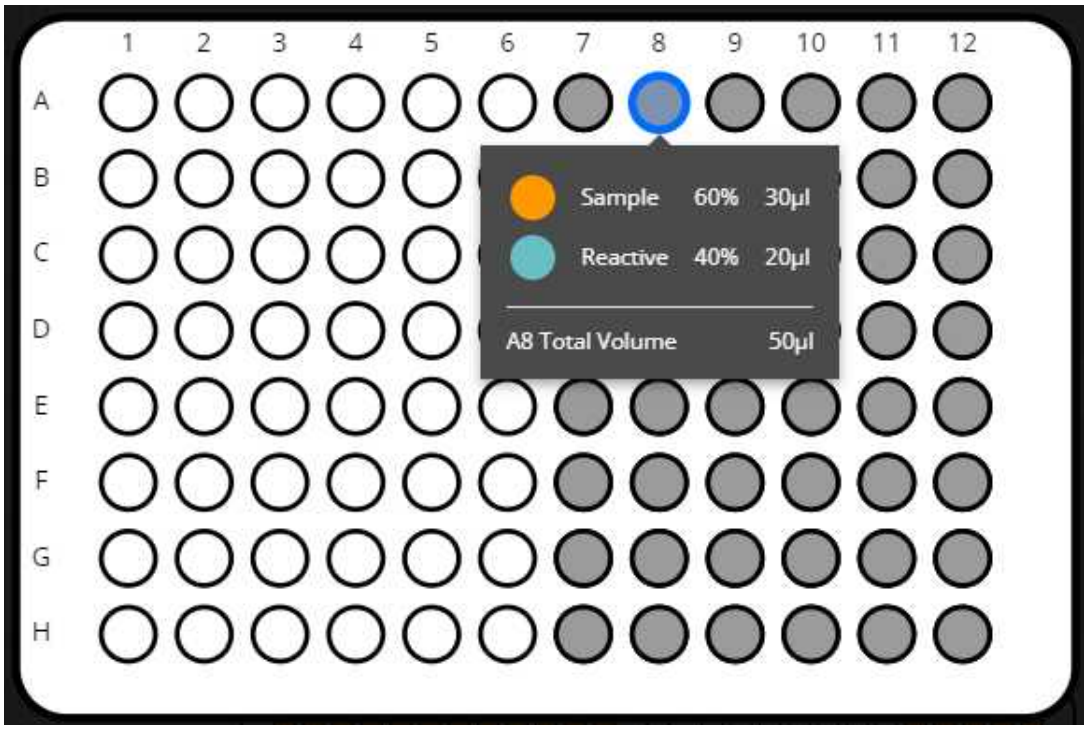
7.7 **Clean platform** of the robot that we are going to perform the protocol 2m

7.8 **Prepare all reagents and labware** in the places the App is showing and take into account the notes in step go to step #3.2 Notes 5m

7.9 **Start run** 35m

Expected result



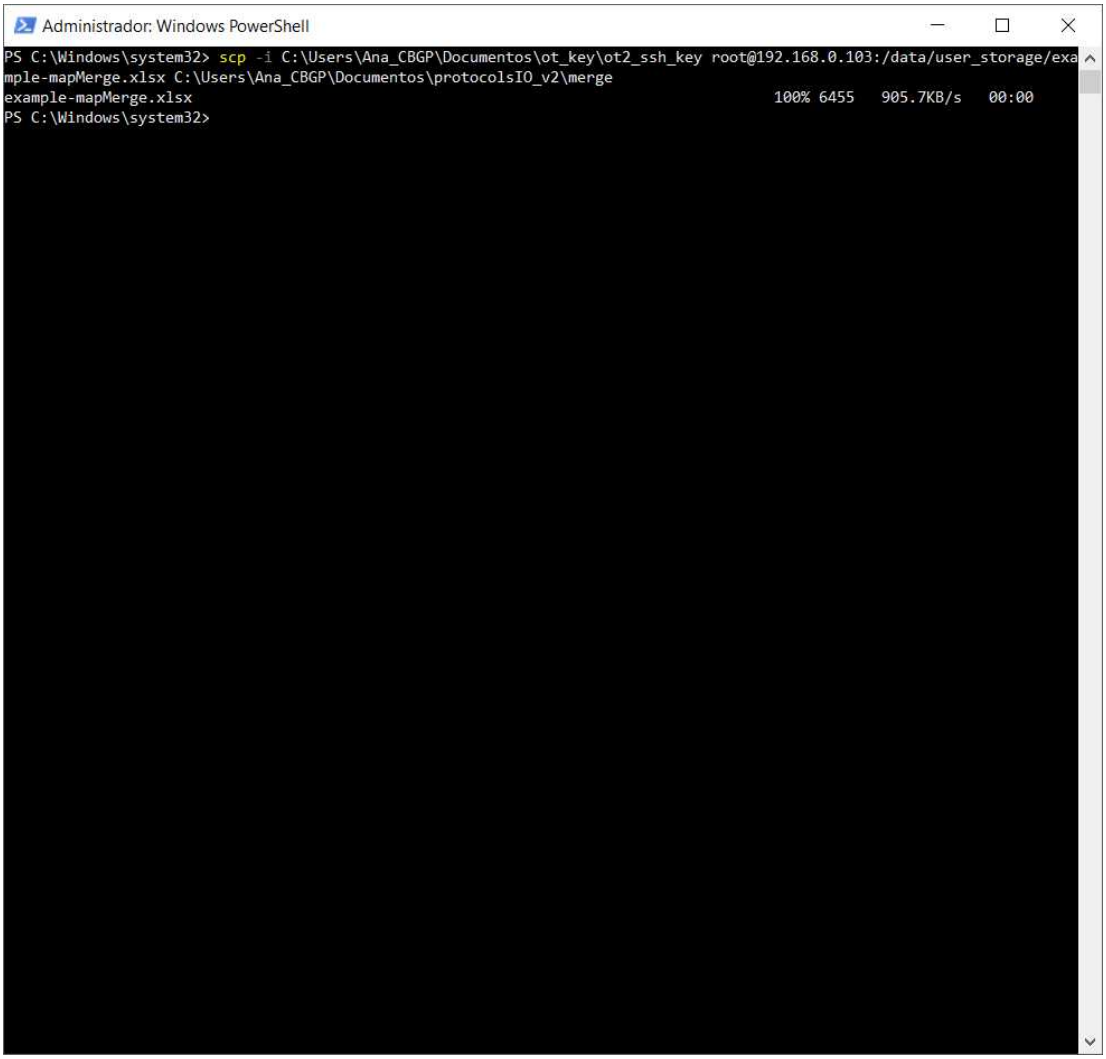


Example of the content of A8 in the labware *Final Plate 1 Slot 6*

Here we will obtain the mix between the volume of the reactive, in this case, water, and the samples selected by the values in the variable file. These positions are seen in the image by the grey wells, and we can see the info of the plate and reactive in the plate on slot 6 in the second picture

7.10 **Retrieve labwares from the OT** 5m

7.11 **Retrieve the final maps**, in this case, the map file will be called *example-mapMerge.xlsx* (name that is stated in the variable file in the variable *Name File Final Map*) 1m



Command line windows with the transfer command of the final plate(s) map(s) file from the OT to our computer

example-mapMerge.xlsx 9KB

For more information about how to retrieve files from the robot Linked Protocol

Protocol references

pBLAM1-x: standardized transposon tools for high-throughput screening *(Synthetic Biology)* https://doi.org/10.1093/synbio/ysad012

The Laboratory Automation Protocol (LAP) Format and Repository: A Platform for Enhancing Workflow Efficiency in Synthetic Biology (*ACS Synth. Biol.*) https://doi.org/10.1021/acssynbio.3c00397