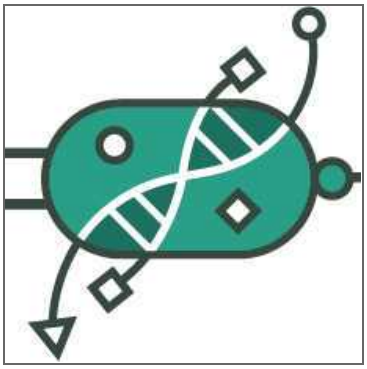


VERSION 2

DEC 22, 2023

OPEN ACCESS

DOI:
dx.doi.org/10.17504/protocols.io.6qpvr4o62gmk/v2

Protocol Citation: Ana Mariya Anhel, Lorea Alejaldre, Ángel Goñi-Moreno 2023. OT-2 Protocol to transfer volume from several plates to a single plate . **protocols.io** https://dx.doi.org/10.17504/protocols.io.6qpvr4o62gmk/v2 Version created by Lorea Alejaldre

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

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

[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

Ángel Goñi-Moreno: angel.goni@upm.es

biocomp.cbgp

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-OT2-1.0.0**

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

Created: Nov 20, 2023

Last Modified: Dec 22, 2023

PROTOCOL integer ID: 91172

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

GUIDELINES

This protocol was developed with python 3.7.1, OT App Software Version 6.3.1 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 6.3.1
- 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 Filter Tip Rack 20 µL	NAME
Tip rack	TYPE
Opentrons	BRAND
-	SKU
https://labware.opentrons.com/opentrons_96_filtertiprack_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 INSTRUCTIONS

It is important to annotate 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 and if 2 samples, within a plate or between different source plates has 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*

Template-VariablesMergeSamples.xlsx 15KB
MergeSamplesInstructions.pdf 202KB

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

2 Transferring *VariablesMergeSamples.xlsx* file to Robot

Transfer the *VariablesMergeSamples.xlsx* to the directory */data/user_storage* of the OT robot that we are going to use to perform the protocol.

Note

Before transferring any file to the OT, we need to know the **IP of the robot**.

This can be obtained in the Networking section of the Device that we will use.
To obtain this info go to **OT-App -> Devices -> Chosen Robot (three dots) -> Robot Settings -> Networking**

In this tab, you can see 2 types of IP; one is shown if both the robot and you are connected to the same Wifi, and the other is shown if the computer and the robot are connected via USB. Both connections can be used for this step

Note

To connect to the robot an **OT-key** should have been previously generated, and it is done with the *ssh-keygen* command and **transferring the public key to the OT.**

For more information about how to generate and set the connection between your computer and the Opentrons robot, visit https://support.opentrons.com/s/article/Setting-up-SSH-access-to-your-OT-2

Here, we present a summary of how to transfer the files in 3 Operative Systems: *Windows, MacOS* and *Linux*

MacOS/Linux
We will use the command line with scp to transfer the file *VariablesMergeSamples.xlsx* to the OT system.

We need to perform the following command

Command

File passing from our computer to robot's linux (OT raspberry)

```
scp -i [ot_key] [file] root@[IP_OT]:/data/user_storage
```

Note

You could face difficulties transferring files in MacOS Ventura (13) and Sonoma (14). These problems can be solved by adding the argument -O (uppercase o) to the command

Command

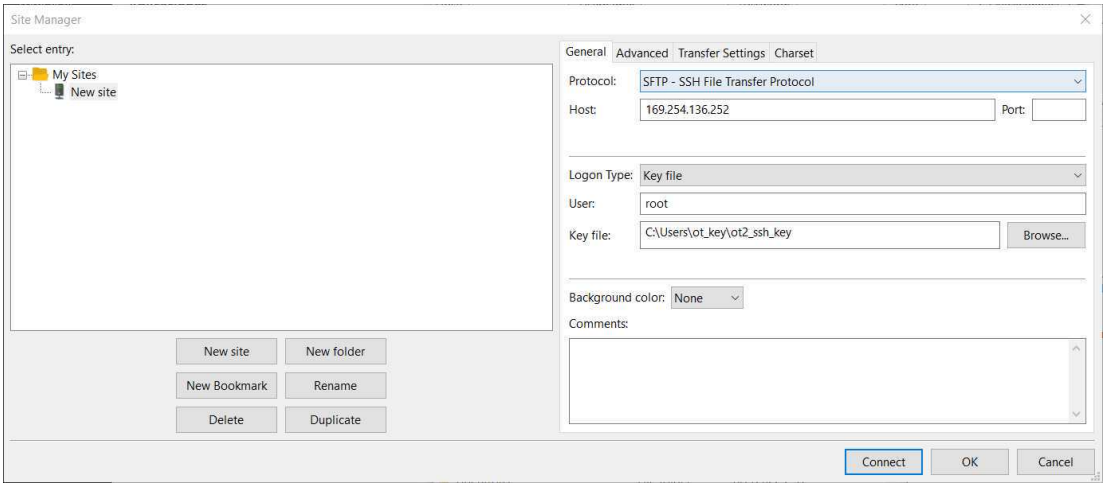
Transferring files to OT (MacOS 13 and 14)

```
scp -Oi [ot_keypath] [file path]  root@[OP_robot]:/data/user_storage
```

Windows
There are several ways to send files from a Windows to a Linux (for example, with a virtual machine or Windows Powershell in the latest versions of Windows).

Here, we will use **FileZilla** (https://filezilla-project.org/download.php?type=client).

Go to **File -> Site Manager -> New Site -> Change Protocol to SFTP**. Then, introduce in Host the OT IP, change the *Logon Type* to key file, change *User* to root and give the directory where the ot key is. It should look something like this



Example of setting the FileZilla to transfer files from Windows (our computer) to Linux (OT)

Then press *Connect*, and we will have a connection between our computer and the robot.

After this connection, we should be able to move the file *VariablesMergeSamples.xlsx* (in our computer) to the directory */data/user_storage* in the robot.

This method can be used as well in some Linux and MacOS

Note

Take into account that the IP of the robot could change, so it is possible that it will be needed to change the host in these connections from time to time.

3 Adding the custom labware

There is only a need to do this step when the labware that you are using is not OT official or is not included in the OT app

3.1 *Creation of .json file*

The description file can be obtained by describing the labware dimensions at https://labware.opentrons.com/create/

3.2 *Uploading files to the OT App*

In the OT app, we need to perform the following route: **Labware -> Import -> Choose File -> Select file we have created in step 3.1**

Expected result

After uploading the labware you should be able to see the new labware in the Labware tab of the OT App, all custom labware can be found more easily in the category *Custom Labware*

3.3 *Transfer labware files to the robot*

If you are using the entry **LAP-NplateMerging-OT2-1.0.0 and custom labware,** an additional step is needed, which is transferring a folder with the custom labware

We need to create for our custom labware a folder with the API name containing the description file (.json) called 1.json and then transfer that folder to the robot's folder */data/labware/v2/custom_definitions/custom_beta* in a similar way as in the Step 2 but with the difference that is a directory that needs to be transferred and not a file.

Command

Transferring the custom labware to OT (Linux)

```
scp -i [ot_key] -r [directory_custom_labware] root@[IP_OT]:/data/labware/v2/custom_definitions/custom_beta
```

Note

We do not need to execute this part every time the protocol is used, only when that labware is not included in the OT official labware and these directories are not in the robot

Prepare RobotOS

4 Install needed packages

This script needs the package *openpyxl*, which is not installed by default in the OT-2 robots

Note

This step is only needed if the package is not installed in the robot, not every run of the protocol

If the package is not installed, an error will appear when running the script in the robot. While simulating the script in the app, this error will appear, but you can ignore it

4.1 *Connect to the robot*

go to step #2 to find the IP of the robot in which you want to run the script

To connect to the robot, you can do it via ssh with the following command

Command

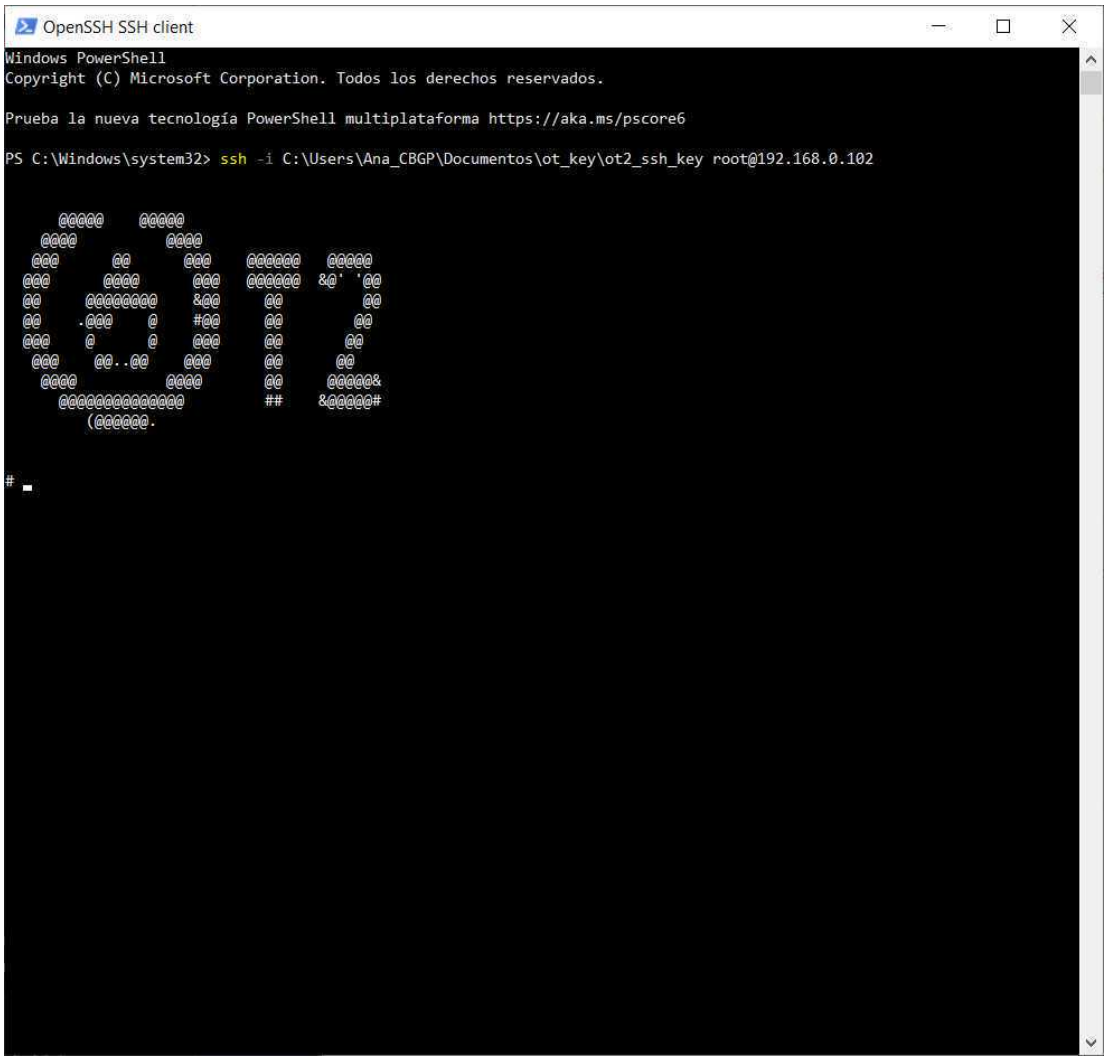
Connect to Linux based OT via ssh

```
ssh -i [path ot_key] root@[Robot_IP]
```

In Windows you can do this command in Windows Powershell

Expected result

If the connection has been successful you should obtain a screen similar to the following image



Drawing obtained when entering an OT-2 system

4.2 *Install the package*

Once inside the robot's system, you need to run the following command

Command

Install openpyxl package (Linux 4.14.74-v7)

pip install openpyxl

Note

For more information about installing packages in the opentrons robots, check the following Opentrons page: https://support.opentrons.com/s/article/Using-Python-packages-in-Python-API-protocols

Running Protocol

5 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 with version 6.3.1 of the OT-App and has been tested until version 7.0.2

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/

5.1 *Load the script in the App*

Protocols -> Import -> Drag Python script

Note

The last script version can be found at https://github.com/BiocomputationLab/LAPrepository/tree/main/LAPEntries (the name of this file is the user's choice). 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.cbgp.upm.es/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 6.3.1 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

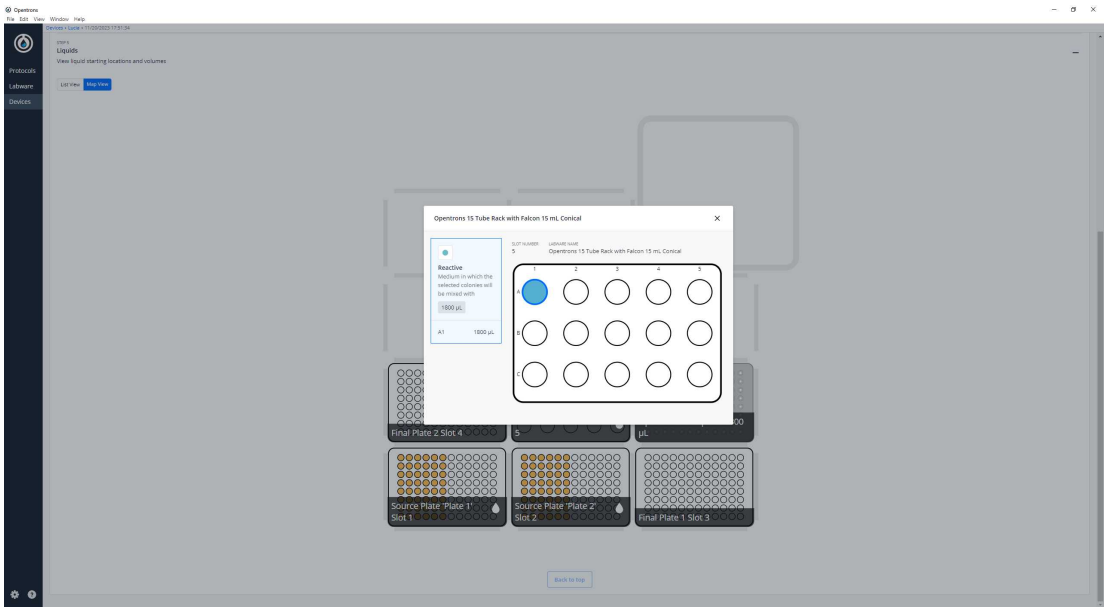
5.2 *Select Robot to Perform Script*

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* tab and in the *Liquids* tab, the reagents and their corresponding volume.

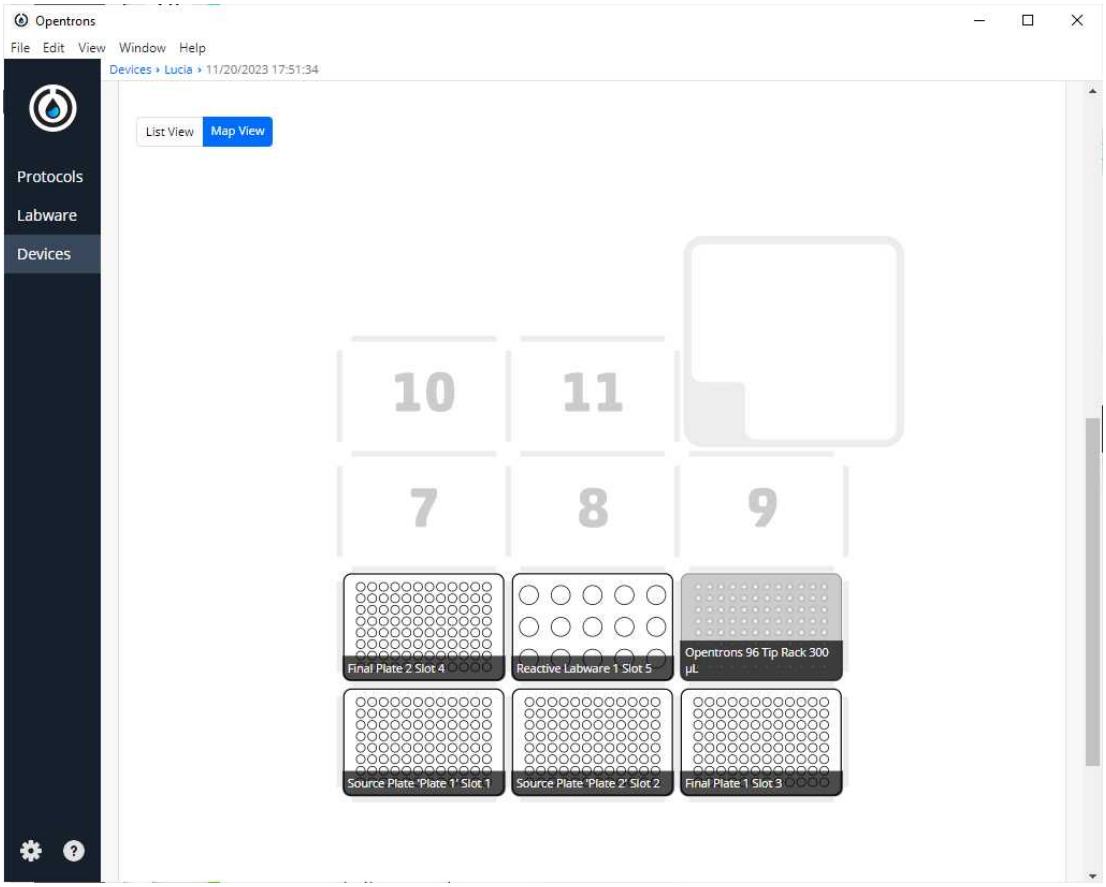
Expected result

A **liquid setup** should look like the following image, where you can find the samples in the initial plates and the reactives in the Falcon labware



Liquid Set-up Example of a Samples Merge Protocol

A **labware setup** should look like the following image, where you can find the initial and final plates, the falcon labware (if needed) to store the reagent and the corresponding tip rack(s)



Labware Set-up example of a Samples Merge Protocol

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).

6 Run Protocol in OT

6.1 *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

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

6.3 *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

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

6.5 *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

7 Retrieve labware from the OT

8 Import map from robot

There will be a map with the name set in the variable *Name File Final Map* set 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 and reproduce it by transferring the files from the robot to the computer.

Command

Transferring files from OT to computer (Linux, macOS)

```
scp -i [path_ot_key] root@[IP_robot]:/data/user_storage/[name_map].xlsx [ultimate_path_computer]
```

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

9 **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

9.1 Prepare variable file

10m

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

VariablesMergeSamples.xlsx22KB

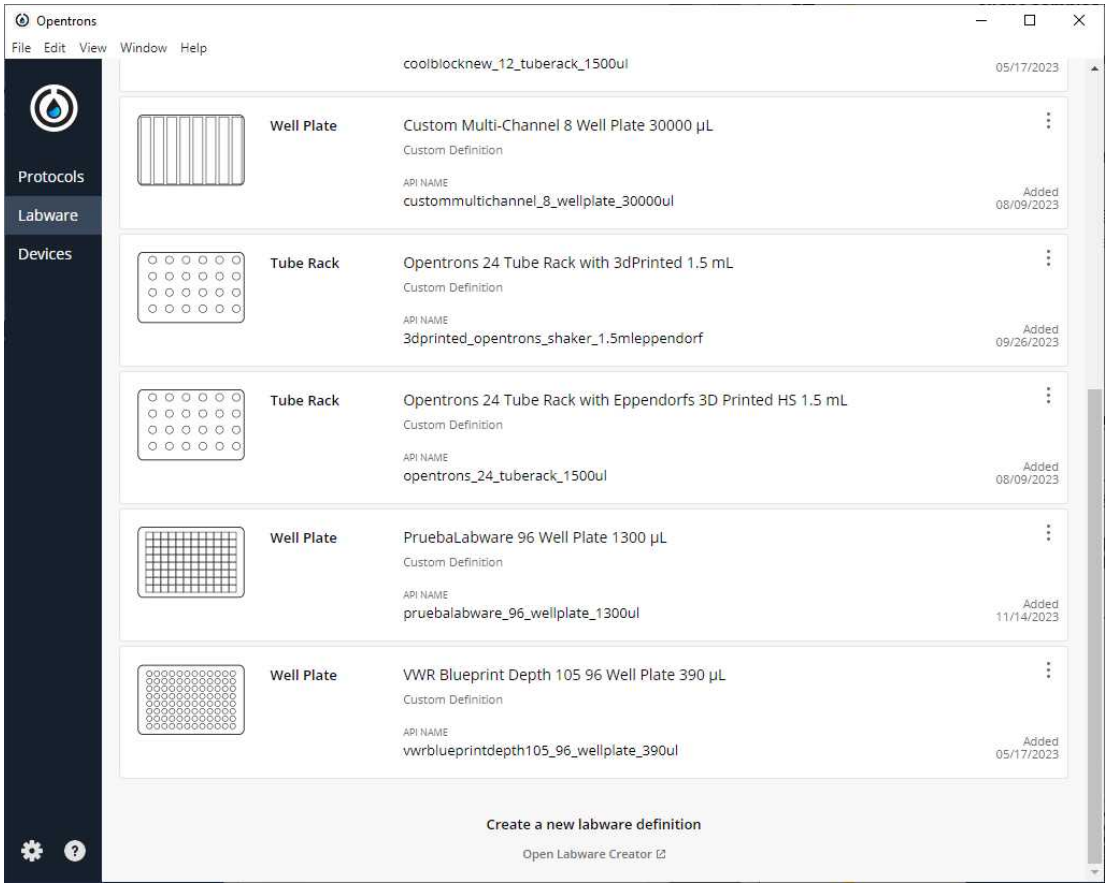
9.2 Upload custom labware to app

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/)

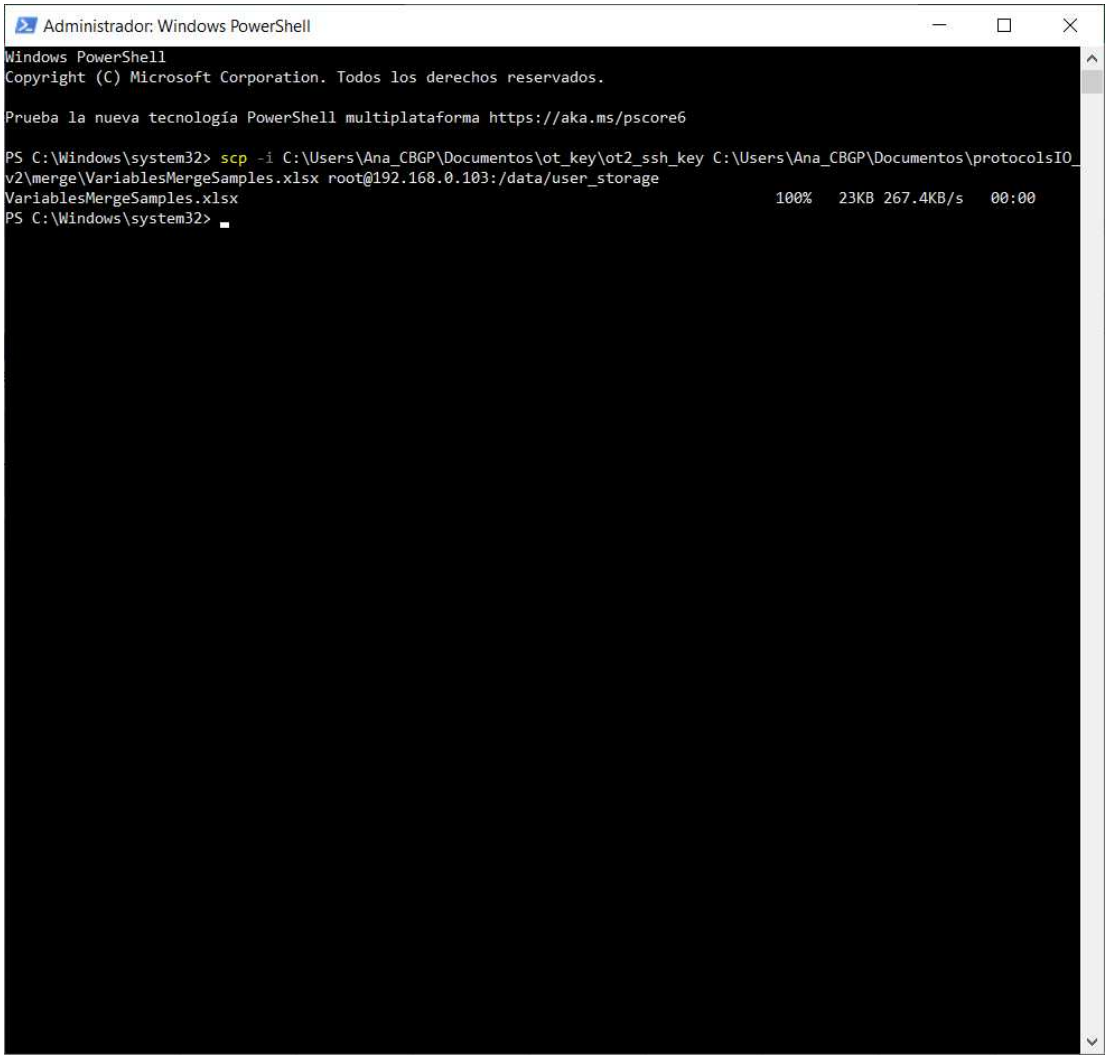
vwrblueprintdepth105_96_wellplate_390ul.json11KB

Upload it to the labware and make sure it is loaded in the app



List of custom labware recorded in the Opentrons App

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



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

9.4 Because we are using version 1.0.0 of the script in this example, we will transfer the directory of th 1m labware as well (here we have attached a zip, but it is the folder that must be transferred, not the zip)

vwrblueprintdepth105_96_wellplate_390ul.zip1KB

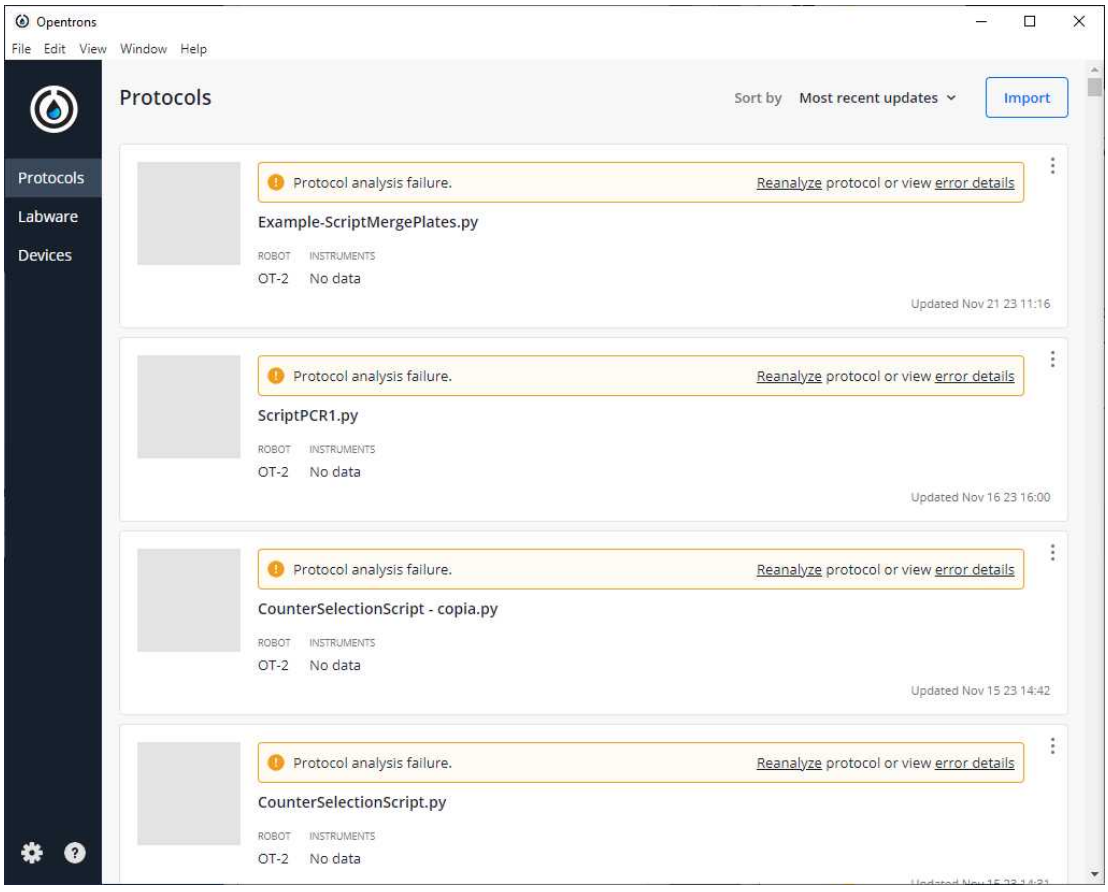
Command

Transferring the used custom labware to OT (Linux)

scp -i [ot_key] -r vwrblueprintdepth105_96_wellplate_390ul root@[IP_OT]:/data/labware/v2/custom_definitions/custom_beta

9.5 Import the script that we have downloaded from the step go to step #5.1 (I named it *Examp* 30s *ScriptMergePlates.py*) to the OT-App

Example-ScriptMergePlates.py 47KB



Result of importing the Python script into the OT-App

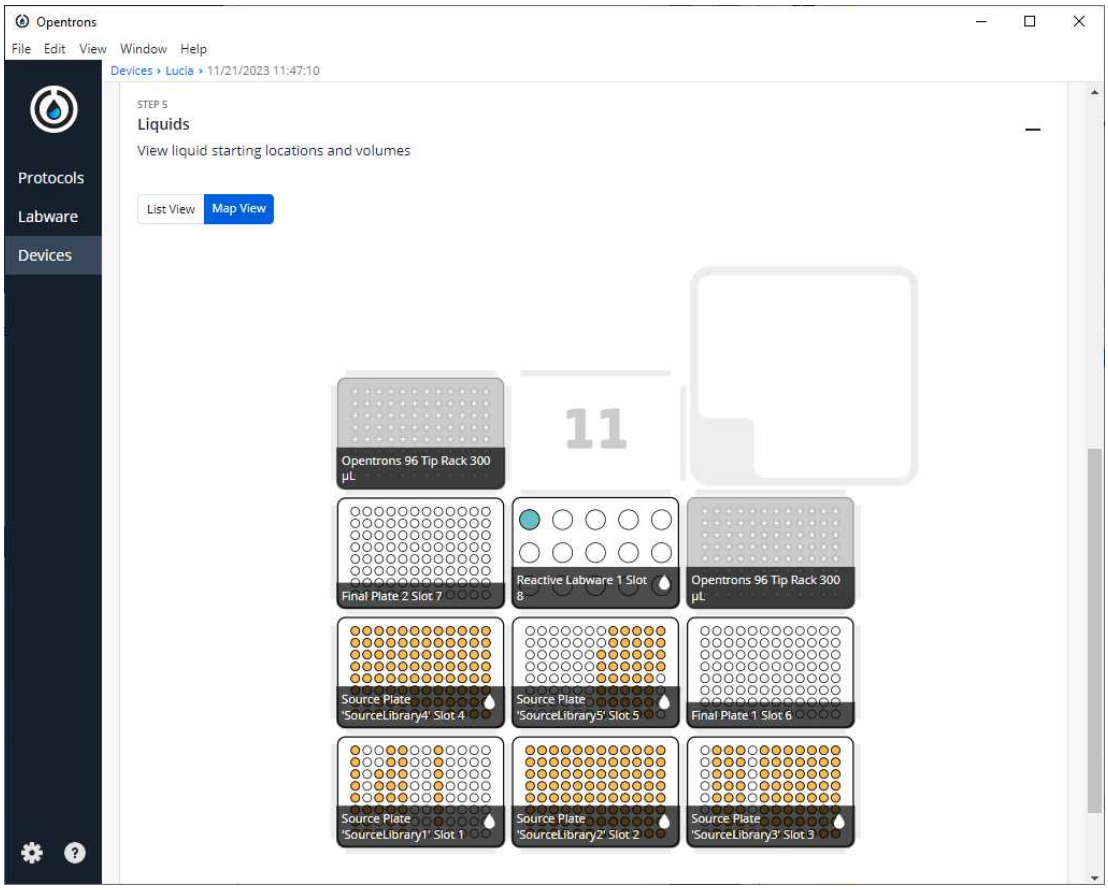
As we can see, we have an error, but that is programmed because the script is meant to work in the robot but not in your computer

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

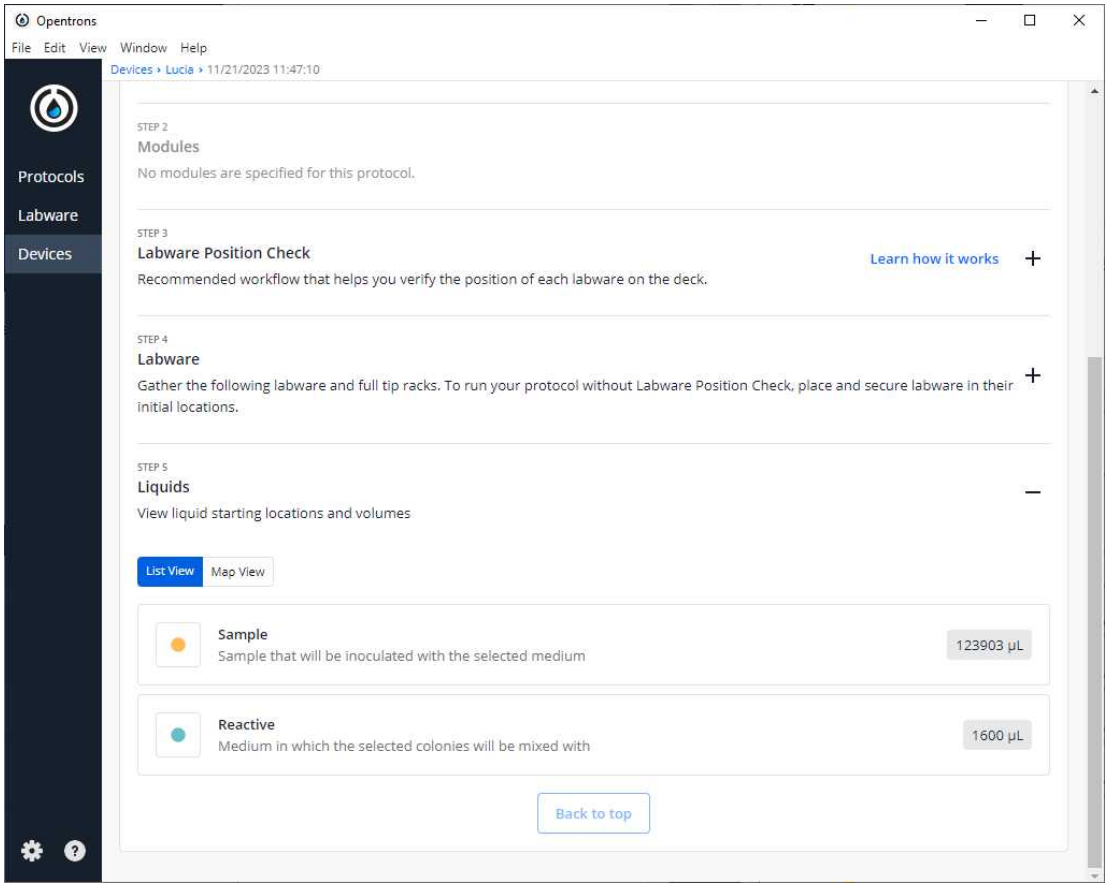
1m

***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

9.7 Turn the HEPA filter module 30s

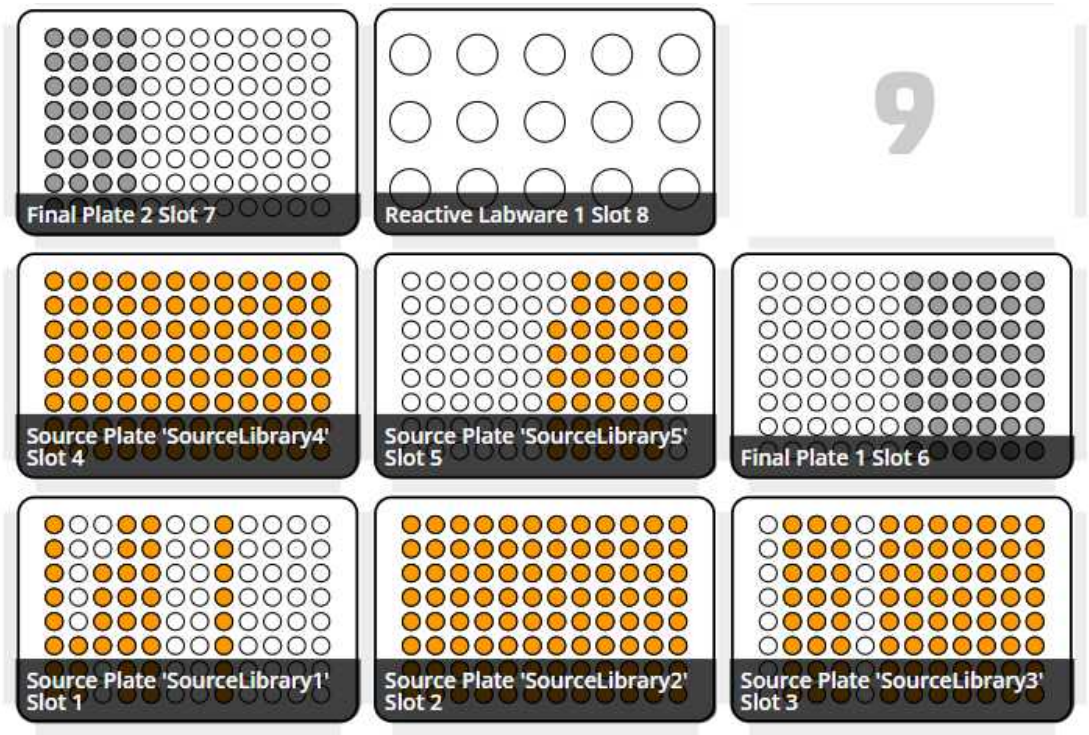
9.8 Clean platform of the robot that we are going to perform the protocol 2m

9.9 Prepare all reagents and labware in the places the App is showing and take into account the notes 5m

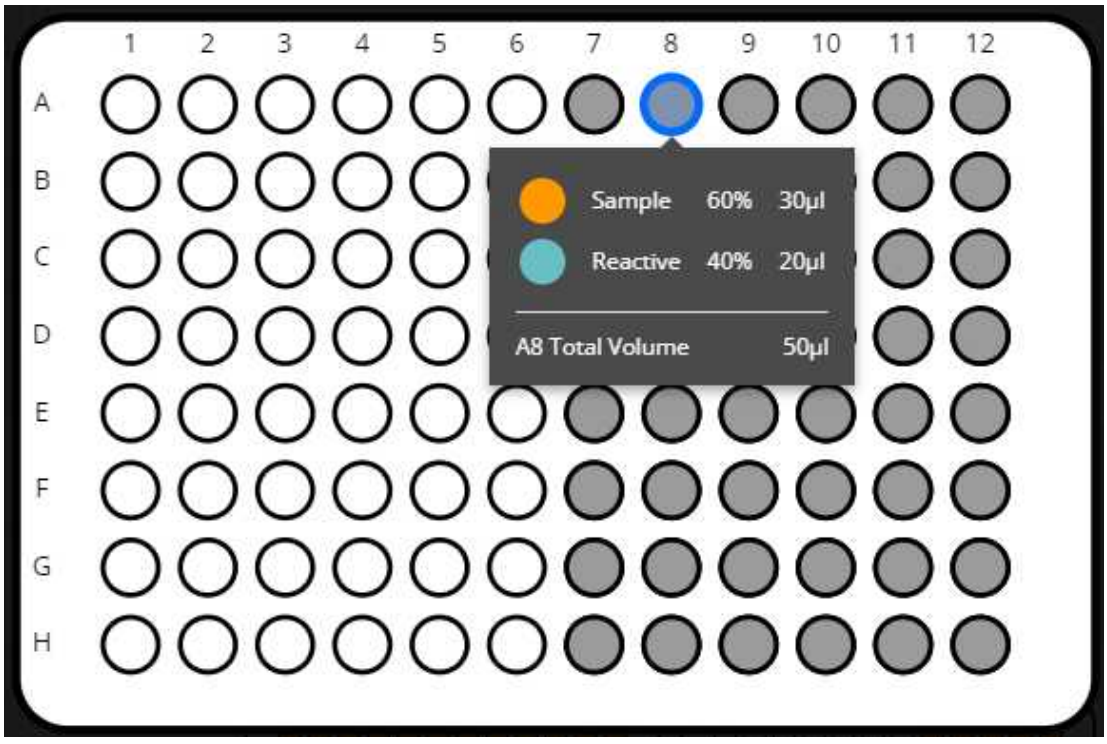
step go to step #5.2 Notes

9.10 Start run 35m

Expected result



Final layout of source and final plates in the run



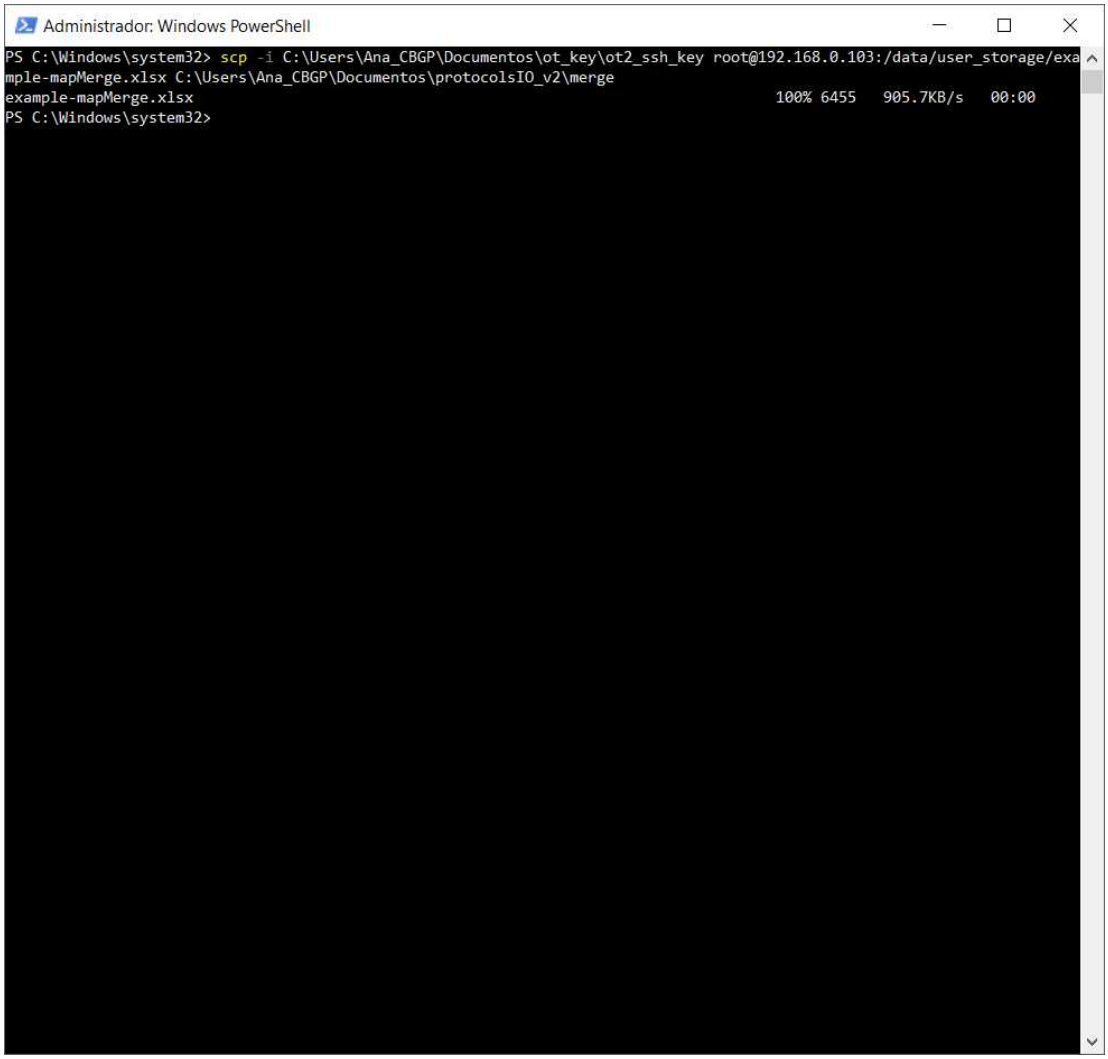
Example of the content of A8 in the labware *Final Plate 1 Slot 6*

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

9.11 Retrieve labwares from the OT 5m

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



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

example-mapMerge.xlsx9KB