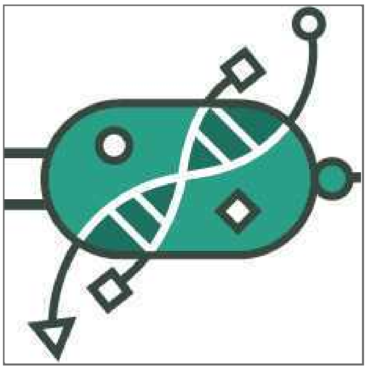


VERSION 3

JAN 16, 2024

OPEN ACCESS

DOI:
dx.doi.org/10.17504/protocols.io.5qpvor5xdv4o/v3

Protocol Citation: Ana Mariya Anhel, Lorea Alejaldre, Ángel Goñi-Moreno 2024. OT-2 Counter-Selection. **protocols.io** https://dx.doi.org/10.17504/protocols.io.5qpvor5xdv4o/v3Version created by Ana Mariya Anhel

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

OT-2 Counter-Selection V.3

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 select the samples from one or more source plate(s) that has 2 different selecting conditions: one condition gives higher values and the other lower values. From these selected samples this protocol create final plates with the samples alone or mixed with different reactive(s).

The instructions for running this protocol are a python script for an Opentrons 2 robot and a excel file with several variables making the protocol modular to reactives, volumes of transfer, type of plates, etc.

In our lab we are selecting samples by their after night OD in 2 different antibiotics and create PCR plates (with water) and 2 stock plates (with glycerol), but it can be used with other selection characteristics such as expression of GFP and with other types of reactives.

This protocol description corresponds to the entry **LAP-ColonyCounterSelection-OT2-1.1.0** of the LAP repository

Created: Jan 16, 2024

Last Modified: Jan 16, 2024

PROTOCOL integer ID: 93580

Funders Acknowledgement:
Comunidad de Madrid
Grant ID: Y2020/TCS-6555,2019-T1/BIO-14053
MCIN/AEI
Grant ID: CEX2020-000999-S,PID2020-117205GA-I00
ERC
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 a counter-selection step in the High-throughput workflow for the genotypic characterization of transposon library variants.
30% glycerol and water have been successfully dispensed using this protocol.

The maximum number of 96-well plates per run given 1 source plates, 1 pipette and replacement of the tiprack set as True is 8 final plates (using 1 tip rack and 1 tube rack for falcons tubes).

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

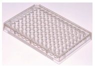
Equipment

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

HEPA Module	NAME
Opentrons	BRAND
OT-2-HEPA	SKU
https://opentrons.com/modules/hepa-module/	LINK

Equipment

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

SAFETY WARNINGS

❗ It is important to use HEPA module to work in sterility

BEFORE START INSTRUCTIONS

Note that this protocol needs 2 values to do a selection for each source plate

Changes respective to previous version:

1. The final given maps of the selected samples is **no longer a CSV** for each source plate but an **XLSX final file** with all the map(s) where each source plate map is 1 sheet of that file
2. Add variable in Sheet 'General Variables' called **'Name Final File Maps'** that is going to establish the name of the final map
3. Name variable 'Final Map Name' in sheet 'PerPlateVariables' is going to define the name of the sheet in the final map(s) excel file
4. Names of the columns in 'PerPlateVariables' are used now as the name of the source plates for both final layout and identifiers in final maps
5. The identifiers in final map(s) of selected colonies is now **[Position of the sample in source plate] [Name of the column]** instead of **[Position of the sample in source plate] from source Plate [number of the plate]**
6. Changes in names of labware in final layout shown in the OT-App

Files Preparation

1 **Preparing Customized Template**

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

Here we attach a template of the variable file with several sheets and a PDF file explaining each variable:

1. **GeneralVariables:** variables related mainly to the labware that is going to be used
2. **PipetteVariables:** variables related to the pipettes that are going to be used
3. **PerPlateVariables:** variables related to the specifications of each source plate
4. **Sheet(s) for comparable values (OD, fluorescence, etc):** values that will be compared against the threshold established in the PerPlateVariables sheet -> *Not in the template but should be present in the final file*

Template-VariablesCounterSelection.xlsx

CounterSelectionInstructionsv110.pdf

1.1 *Fill the template with the corresponding values*

1.2 *Store it with the name VariablesCounterSelection.xlsx*

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

2 Transferring file to Robot

Transfer the *VariablesCounterSelection.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 in **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 *VariablesCounterSelection.xlsx* to the OT system.

We need to perform the following command

Command

File passing from linux (our computer) to linux (OT raspberry)

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

Note

You can 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

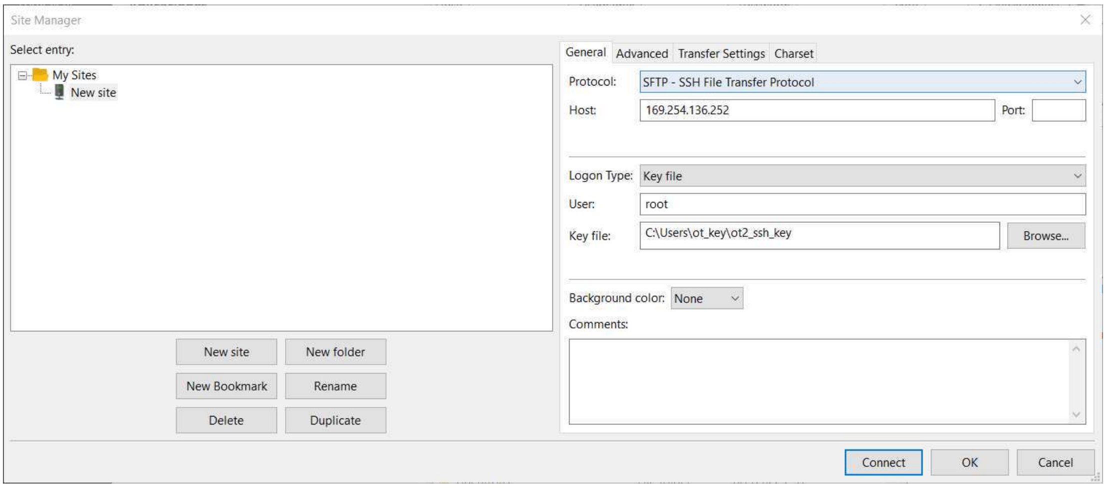
Transferring files to OT (MacOS 13 and 14)

```
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



Protocol: SFTP - SSH File Transfer Protocol
Host: 169.254.136.252
Logon Type: Key file
User: root
Key file: C:\Users\ot_key\ot2_ssh_key
Background color: None

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 *VariablesCounterSelection.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 necessary 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 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 file 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 robot*

If you are using the entry **LAP-ColonyCounterSelection-OT2-1.1.0 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 Robot OS

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 in every run of the protocol.

If the package is not installed, an error will appear when running the script in the robot. This error will appear while simulating the script in the app, 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

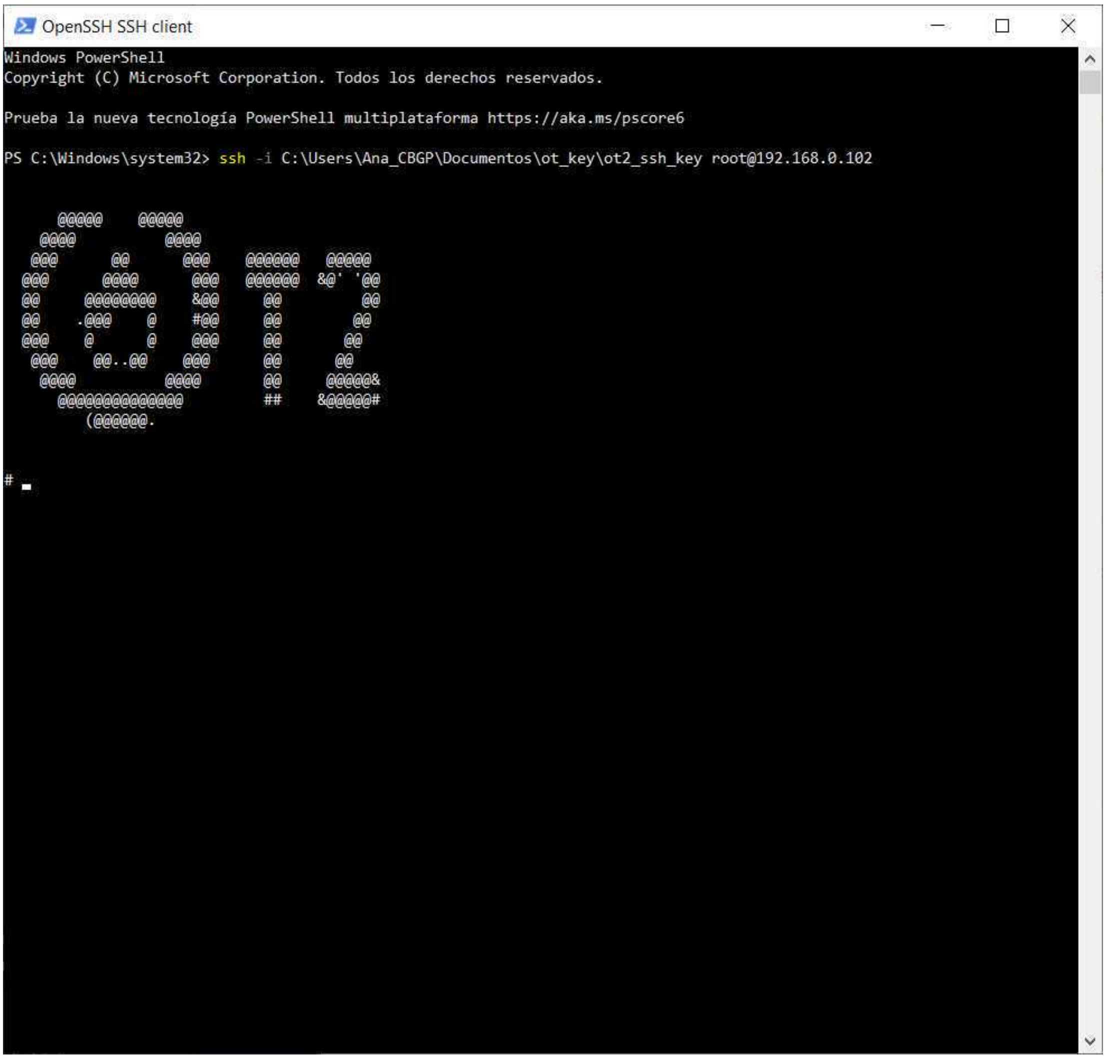
Connect to Linux based OT via ssh

```
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



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Windows\system32> ssh -i C:\Users\Ana_CBGP\Documentos\ot_key\ot2_ssh_key root@192.168.0.102


      @@@@@    @@@@@
    @@@@          @@@@
   @@@      @@      @@@    @@@@@@   @@@@@
  @@@      @@@@      @@@   @@@@@@  &@' '@@
  @@     @@@@@@@@    &@@     @@         @@
  @@    .@@@    @    #@@     @@        @@
  @@@    @      @    @@@     @@       @@
   @@@    @@..@@    @@@      @@      @@
    @@@@          @@@@       @@     @@@@@&
      @@@@@@@@@@@@@@         ##    &@@@@@#
         (@@@@@@.


# 
```

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 load 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 it has been tested up until the version 7.0.2

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

Software

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

5.1 *Load the script in the OT 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-ColonyCounterSelection-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

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

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*

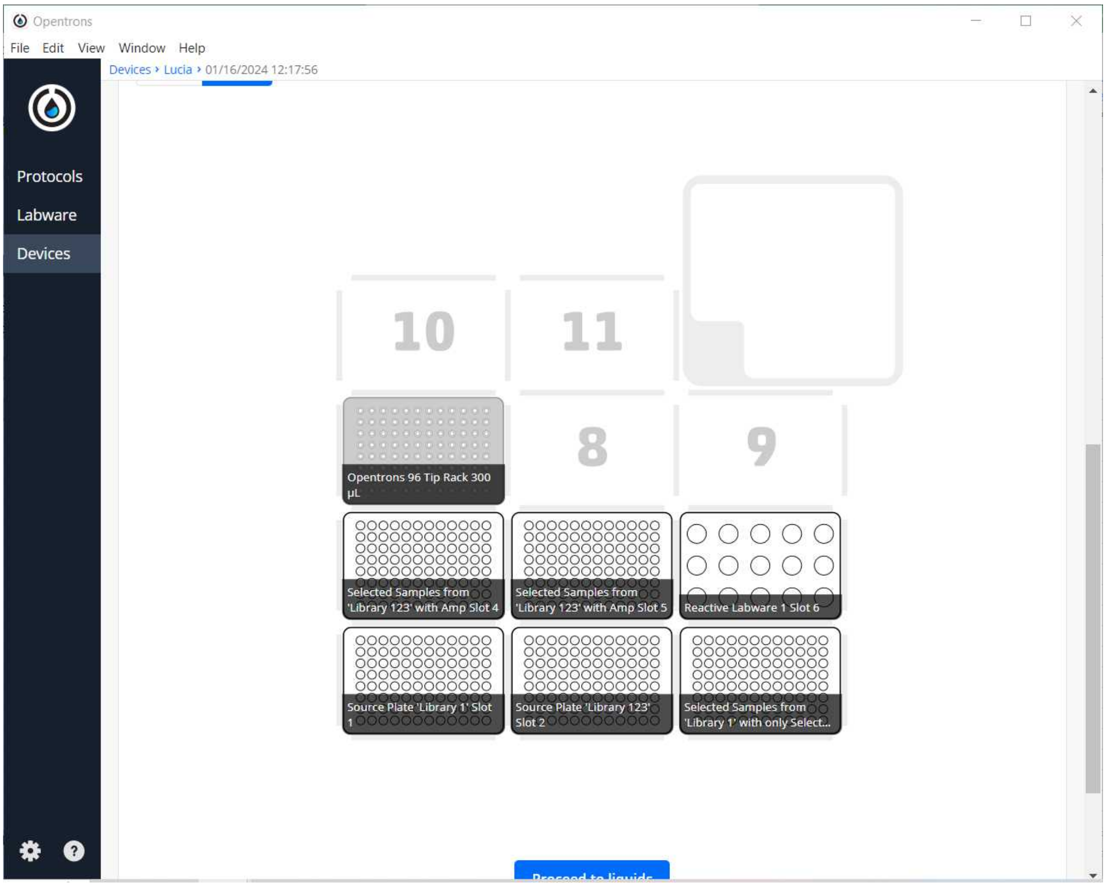
Click in the protocol -> Start setup -> Choose the OT where the file *VariablesCounterSelection.xlsx* is -> Proceed To Setup

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

In case the protocol with the set variables cannot run, an error will occur during the run. A lot of errors are contemplated already and have a specific message that gives a hint of what could have gone wrong.

Expected result

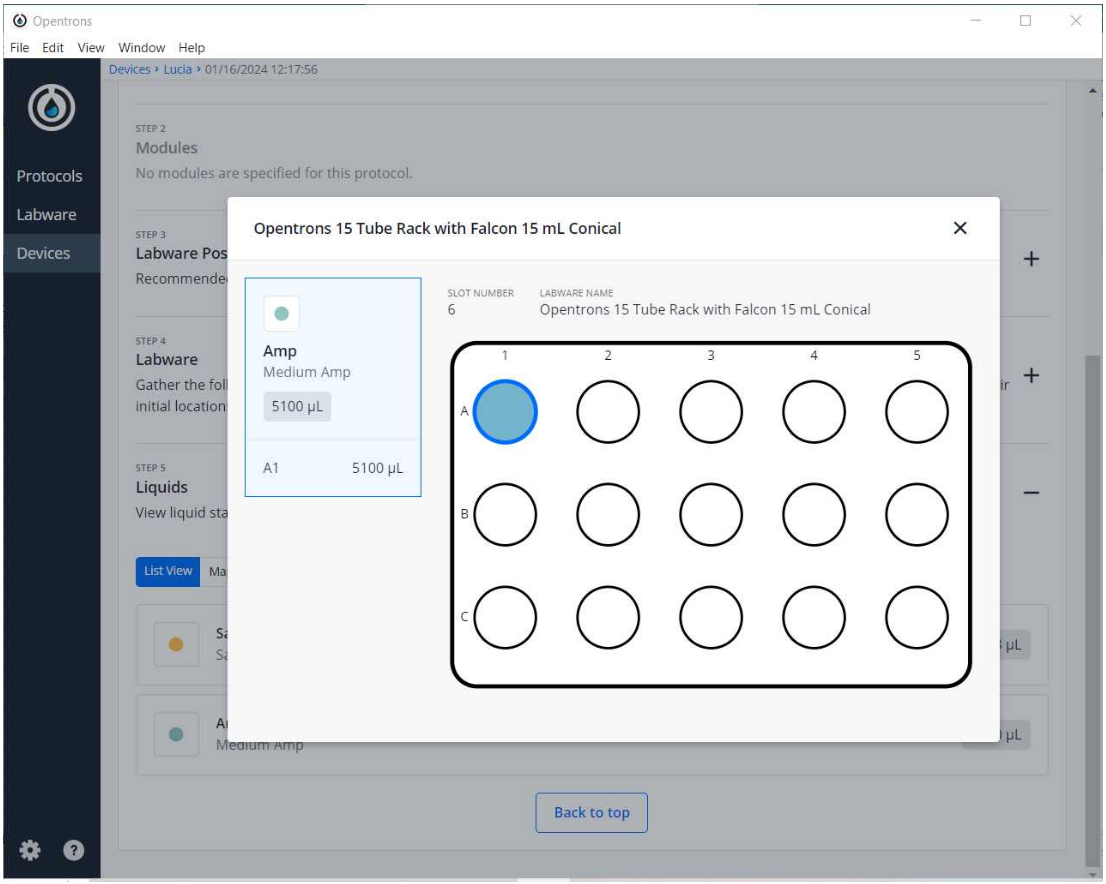
A labware setup should look like the following image, where you can find the initial and final plates, the Falcon labware to store the reagents and the corresponding tips


Opentrons
File Edit View Window Help
Devices › Lucia › 01/16/2024 12:17:56
Protocols
Labware
Devices
10
11
8
9
Opentrons 96 Tip Rack 300 µL
Selected Samples from 'Library 123' with Amp Slot 4
Selected Samples from 'Library 123' with Amp Slot 5
Reactive Labware 1 Slot 6
Source Plate 'Library 1' Slot 1
Source Plate 'Library 123' Slot 2
Selected Samples from 'Library 1' with only Select...
Proceed to liquids

Labware Set-up example of Counter Selection Protocol

Expected result

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



STEP 2
Modules
No modules are specified for this protocol.

STEP 3
Labware Pos
Recommende

STEP 4
Labware
Gather the foll
initial location

STEP 5
Liquids
View liquid sta

Opentrons 15 Tube Rack with Falcon 15 mL Conical

Amp
Medium Amp
5100 µL

A1 5100 µL

SLOT NUMBER 6

LABWARE NAME Opentrons 15 Tube Rack with Falcon 15 mL Conical

Liquid Set-up Example of Counter Selection 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 all the final plates.**

On the other hand, the volume of the reagents is exactly what is needed, so it is **suggested to pour always more to take into 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 needed)*

✱

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 4 parts:

1. Select colonies that comply with the selected parameters (threshold)
2. Distribute reagents to respective plates (if set in the input file)
3. Distribute to all plates the selected samples
4. Generate identity maps (to be exported in thefollowing steps)

Expected result

Several plates, with different reagents but the same samples in the same order, in addition to a **file with all the map(s) (XLSX file) located in the folder *****/data/user_storage*** that will give the position in these plates with their identifiers (location in the source plate)

This set of results will be given for each source plate the user has provided

After-Running

7 **Retrieve labware from the OT**

8 **Import map(s) from robot**

There will be as many maps as source plates with the following structure of names: [NameFinalFile Maps].*xlsx*

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

All of the maps will be in the directory */data/user_storage*

Expected result

The map(s) contains the identity (position in original plate) of the samples selected in the places that they have been placed/distributed being the name of the column as the name of the plate

Example

1h 53m

9 **We want to select the samples that have an OD_{600nm} higher than 0.5 in the Antibiotic Transposition genome file and have an OD_{600nm} lower than 0.5 in the Antibiotic ampicillin plasmid file in one of the source plates and the same selection in the second one but with the value 0.6**

After selecting the samples, we want to transfer those samples to final plates that will have different media

We will use a computer with a Windows 10 system

9.1 Excel template that we can find go to step #1 filled and saved with the name *VariablesCounterSelection.xlsx* 10m

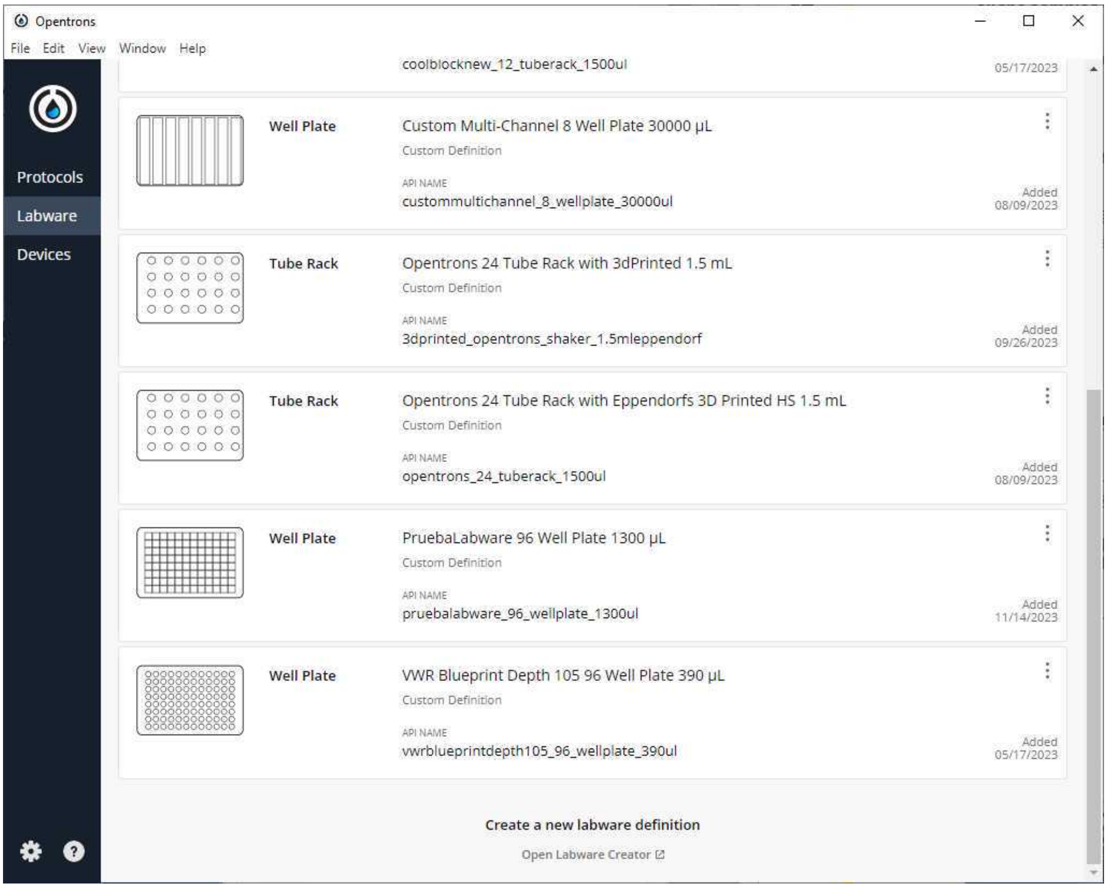
□ VariablesCounterSelection.xlsx

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



coolblocknew_12_tuberack_1500ul 05/17/2023

Well Plate — Custom Multi-Channel 8 Well Plate 30000 µL
Custom Definition
API NAME
custommultichannel_8_wellplate_30000ul
Added 08/09/2023

Tube Rack — Opentrons 24 Tube Rack with 3dPrinted 1.5 mL
Custom Definition
API NAME
3dprinted_opentrons_shaker_1.5mleppendorf
Added 09/26/2023

Tube Rack — Opentrons 24 Tube Rack with Eppendorfs 3D Printed HS 1.5 mL
Custom Definition
API NAME
opentrons_24_tuberack_1500ul
Added 08/09/2023

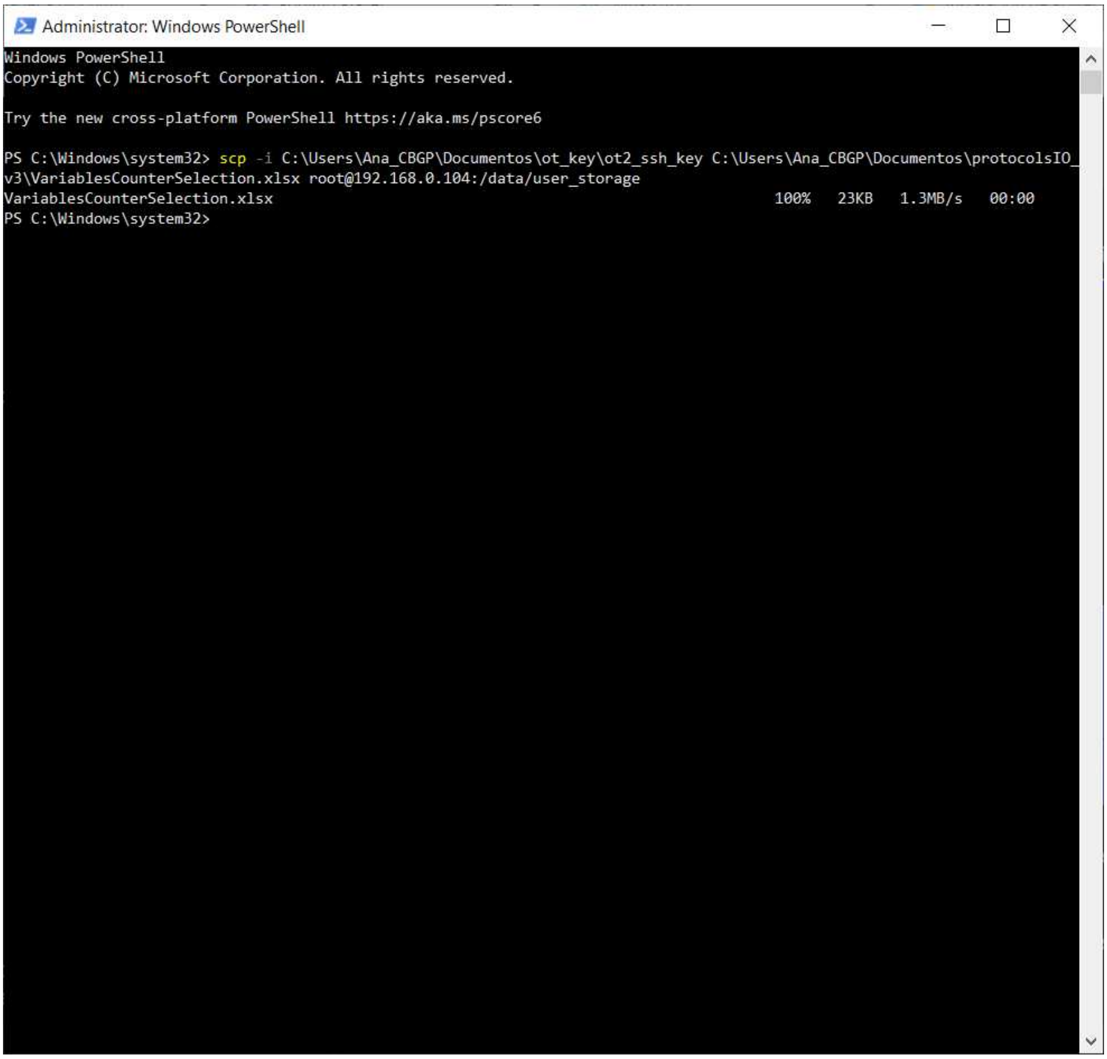
Well Plate — PruebaLabware 96 Well Plate 1300 µL
Custom Definition
API NAME
pruebalabware_96_wellplate_1300ul
Added 11/14/2023

Well Plate — VWR Blueprint Depth 105 96 Well Plate 390 µL
Custom Definition
API NAME
vwrblueprintdepth105_96_wellplate_390ul
Added 05/17/2023

Create a new labware definition

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



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> scp -i C:\Users\Ana_CBGP\Documentos\ot_key\ot2_ssh_key C:\Users\Ana_CBGP\Documentos\protocolsIO_
v3\VariablesCounterSelection.xlsx root@192.168.0.104:/data/user_storage
VariablesCounterSelection.xlsx                                                100%   23KB   1.3MB/s   00:00
PS C:\Windows\system32>
```

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

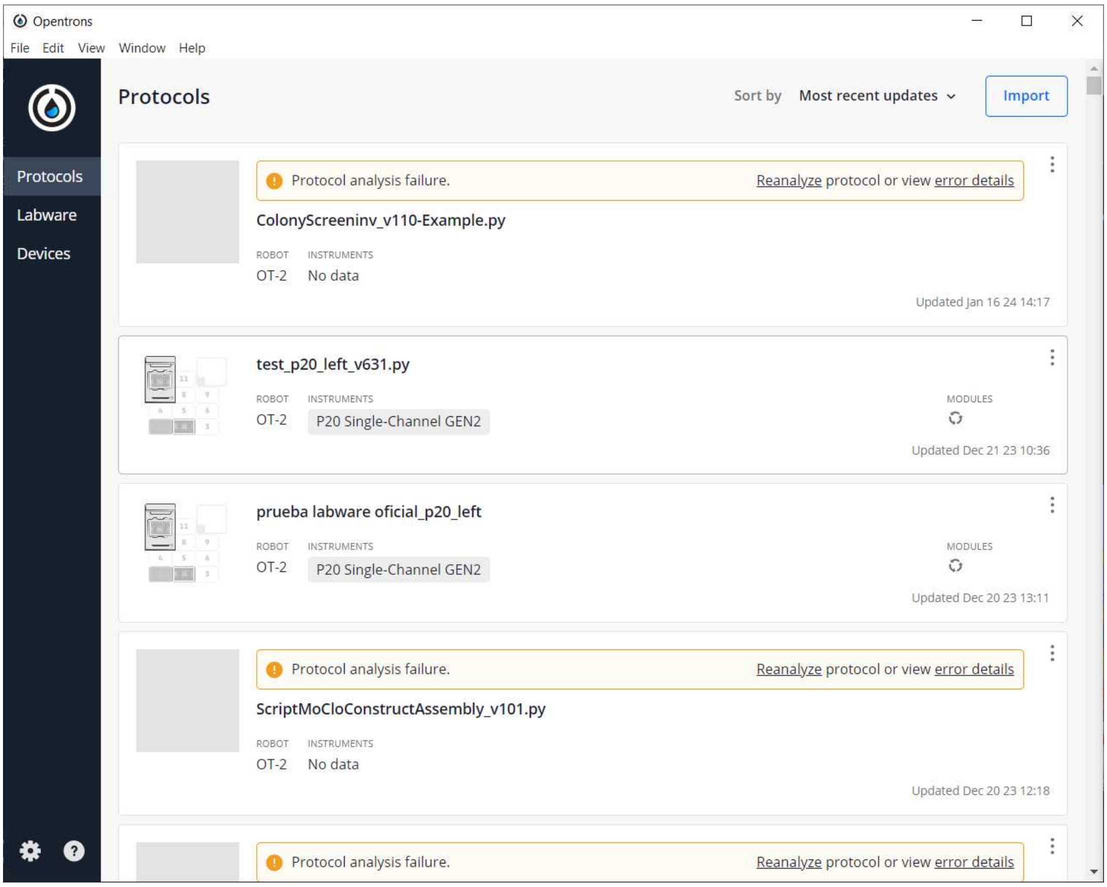
Command

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 (I named it *ColonyScreeninv_v110-Example.py*) to the OT-App 30s

ColonyScreeninv_v110-Example.py



Protocols

Protocol analysis failure. Reanalyze protocol or view error details
ColonyScreeninv_v110-Example.py
OT-2 No data
Updated Jan 16 24 14:17

test_p20_left_v631.py
OT-2 P20 Single-Channel GEN2
Updated Dec 21 23 10:36

prueba labware oficial_p20_left
OT-2 P20 Single-Channel GEN2
Updated Dec 20 23 13:11

Protocol analysis failure. Reanalyze protocol or view error details
ScriptMoCloConstructAssembly_v101.py
OT-2 No data
Updated Dec 20 23 12:18

Protocol analysis failure. Reanalyze protocol or view error details

Result of importing the Python script in the OT-App

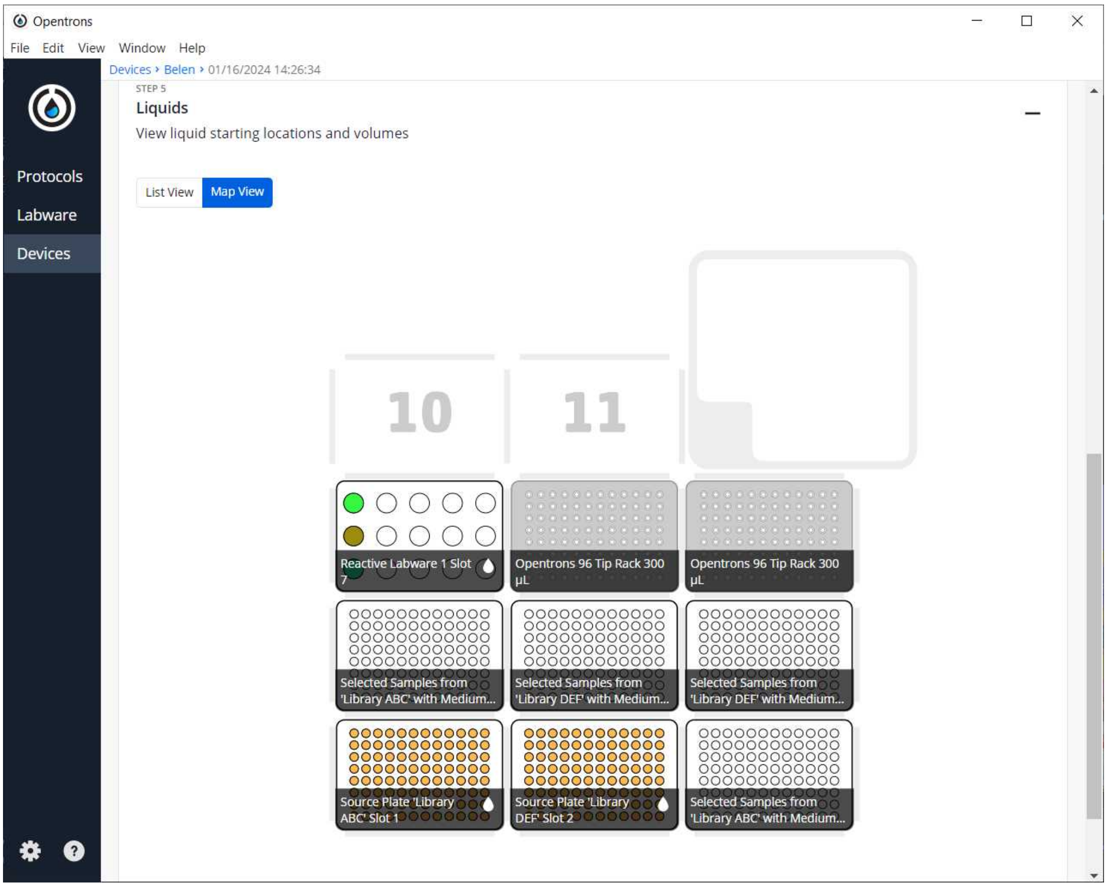
As we can see, we have an error, but that is programmed because the script is meant to work in the robot but not in your computer

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

1m

ColonyScreeninv_v110-Example.py **-> Start setup -> Select 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



Opentrons

File Edit View Window Help

Devices › Belen › 01/16/2024 14:26:34

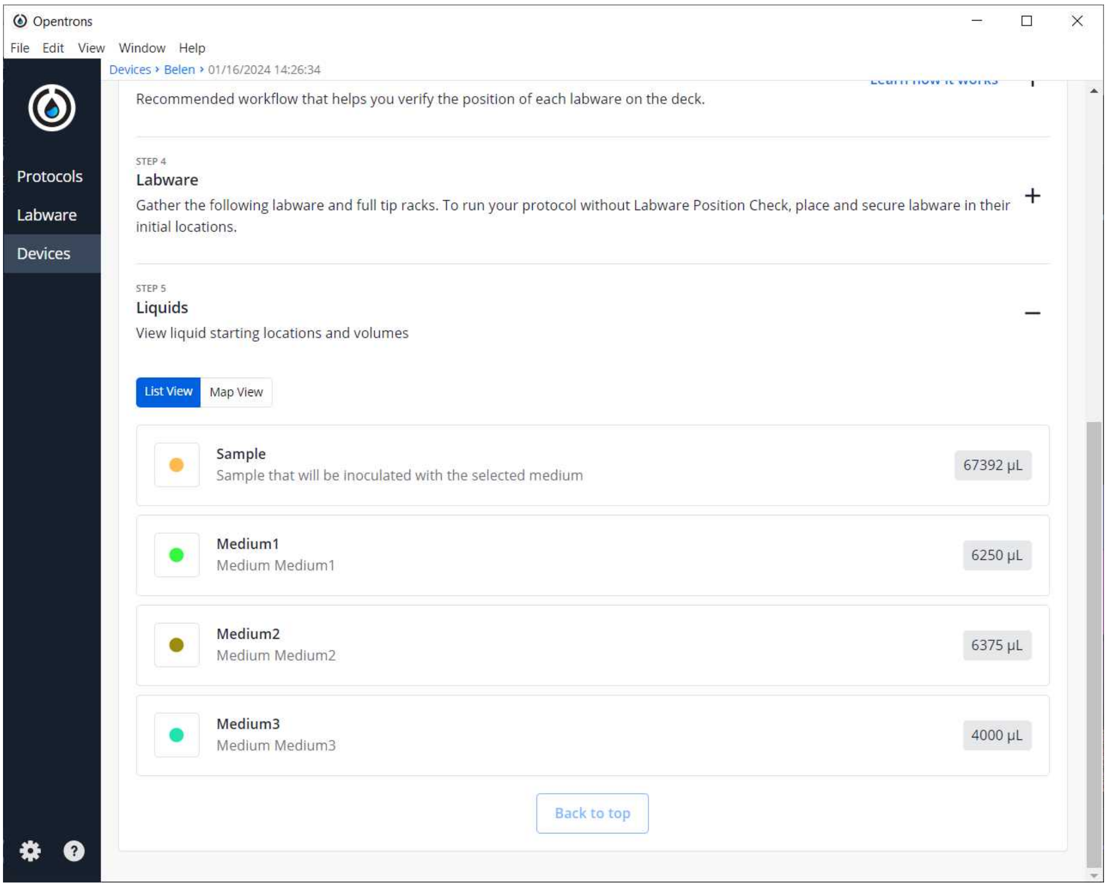
STEP 5

Liquids

View liquid starting locations and volumes

Protocols
Labware
Devices
List View
Map View
10
11
Reactive Labware 1 Slot 7
Opentrons 96 Tip Rack 300 µL
Opentrons 96 Tip Rack 300 µL
Selected Samples from 'Library ABC' with Medium...
Selected Samples from 'Library DEF' with Medium...
Selected Samples from 'Library DEF' with Medium...
Source Plate 'Library ABC' Slot 1
Source Plate 'Library DEF' Slot 2
Selected Samples from 'Library ABC' with Medium...

Labware and liquid set-up layout


Opentrons
File Edit View Window Help
Devices › Belen › 01/16/2024 14:26:34
Protocols
Labware
Devices
List View
Map View
Back to top

Recommended workflow that helps you verify the position of each labware on the deck.

STEP 4
Labware
Gather the following labware and full tip racks. To run your protocol without Labware Position Check, place and secure labware in their initial locations.

STEP 5
Liquids
View liquid starting locations and volumes

●	Sample Sample that will be inoculated with the selected medium	67392 µL
●	Medium1 Medium Medium1	6250 µL
●	Medium2 Medium Medium2	6375 µL
●	Medium3 Medium Medium3	4000 µL

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

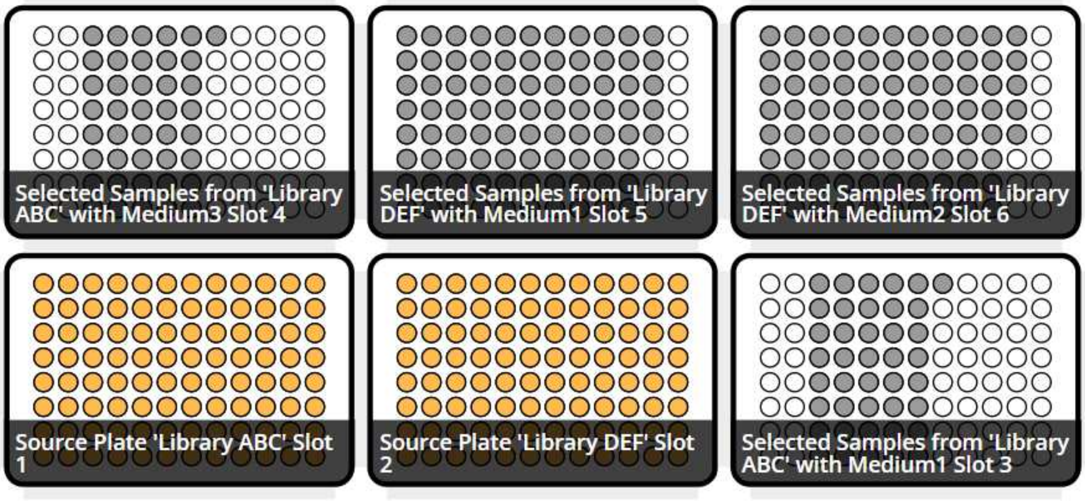
9.9 Prepare all reagents and labware in the places the App is showing and taking into account the note in step go to step #5.2 Notes 5m

9.10 Start run 1h 22m

Final layout of source and final plates in the run

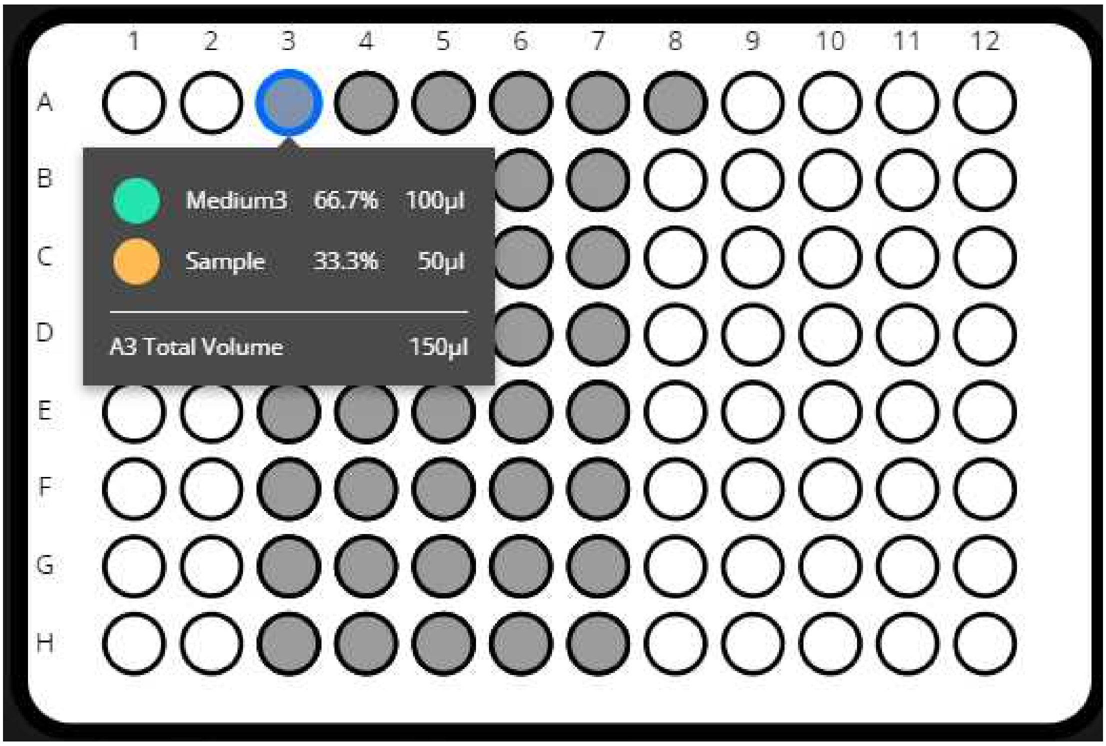
Example of the content of A1 in the labware *Selected Samples from Plate 'Plate 1' with Medium3 Slot 4*

Expected result



Selected Samples from 'Library ABC' with Medium3 Slot 4
Selected Samples from 'Library DEF' with Medium1 Slot 5
Selected Samples from 'Library DEF' with Medium2 Slot 6
Source Plate 'Library ABC' Slot 1
Source Plate 'Library DEF' Slot 2
Selected Samples from 'Library ABC' with Medium1 Slot 3

Final layout of source and final plates in the run



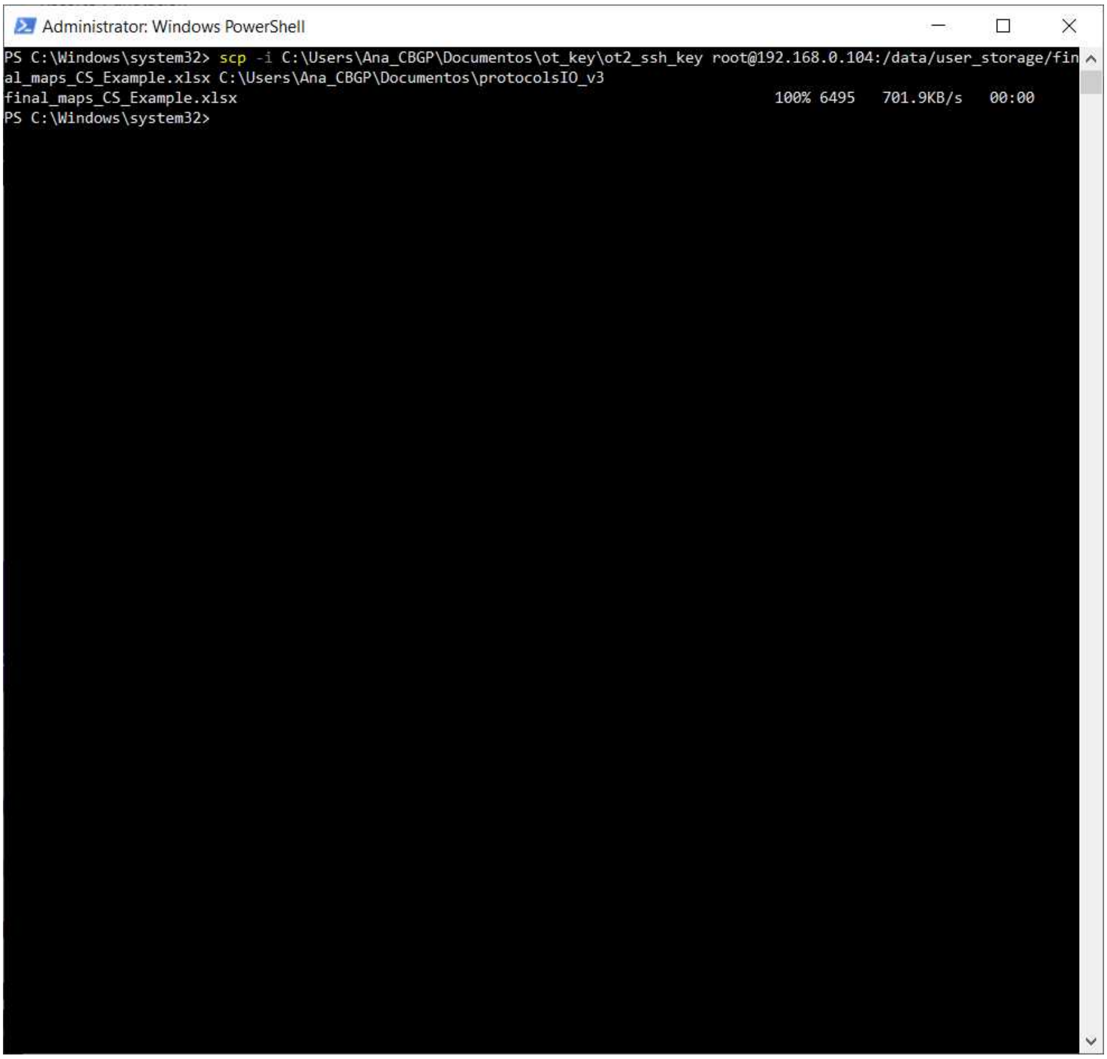
●	Medium3	66.7%	100µl
●	Sample	33.3%	50µl
	A3 Total Volume		150µl

Example of the content of A1 in the labware *Selected Samples from 'Library ABC' with Medium3 Slot 4*

Here, we will obtain the mix between the volume of media and the samples set in the variable file in the final plates. These positions are seen in the image by the grey wells, and we can see the info of the plate and the media in the name of the labware.

9.11 Retrieve labwares from the OT 5m

9.12 Retrieve the final maps, in this case, they will be in a file called *final_maps_CS_Example.xlsx,* of the IDs of the samples that fulfilled the requisites that have the names set in the variable file by us. This file will have 2 sheets, one per source plate, called *mapSelectedColoniesABC* and *mapSelectedColoniesDEF* for the columns *Library ABC* and *Library DEF*, respectively. 2m



```
PS C:\Windows\system32> scp -i C:\Users\Ana_CBGP\Documentos\ot_key\ot2_ssh_key root@192.168.0.104:/data/user_storage/fin
al_maps_CS_Example.xlsx C:\Users\Ana_CBGP\Documentos\protocolsIO_v3
final_maps_CS_Example.xlsx                                                      100% 6495    701.9KB/s   00:00
PS C:\Windows\system32>
```

command line windows with the transfer command of the samples map from the OT to our computer