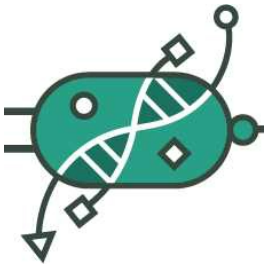
Oct 20, 2025 Version 2

OT-2 Modular Cloning Construct Assembly V.2



DOI
https://dx.doi.org/10.17504/protocols.io.5jyl8p82rg2w/v2

Ana Mariya Anhel[1], Laura Cutugno[1], Ángel oñi-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 oñi-Moreno: angel.goni@cnb.csic.es;

Biocomputation Lab
CNB-CSIC

DOI: https://dx.doi.org/10.17504/protocols.io.5jyl8p82rg2w/v2

Protocol Citation: Ana Mariya Anhel, Laura Cutugno, Ángel oñi-Moreno 2025. OT-2 Modular Cloning Construct Assembly . **protocols.io** https://dx.doi.org/10.17504/protocols.io.5jyl8p82rg2w/v2Version created by **Biocomputation Lab**

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

Created: July 08, 2024

Last Modified: October 20, 2025

Protocol Integer ID: 103008

Keywords: modular cloning construct assembly, modular cloning construct, modular cloning, modular cloning constructs from different part, golden standard modular cloning of level, assembly of the construct, description of the lap repository entry lap, ot2, assembly, protocol in the lap entry link, version of the protocol, set of instruction, plasmid, lap repository, lap repository entry lap, protocol setup, lap format script, protocol, description of robot, temperature profile, laboratory, names of the construct

Funders Acknowledgements:
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 create modular cloning constructs from different parts into a final plate and, optionally, perform the temperature profile needed to the assembly of the constructs.
The output of running this script will be the final plate(s) with the constructs and the mix needed to perform that assembly, and the corresponding map(s) with the names of the constructs in their corrresponding well which will be given by the user in the input file.

This protocol is run by using a **LAP format** script and its corresponding .xlsx file were different customizable variables such as the number of final combinations, volumes of transfer, type of plates, etc...
Specifically, this protocol provides a set of instructions or description of the **LAP repository** entry **LAP-MoCloAssembly-OT2-2.0.0**. You can find the script and complementary information for this specific version of the protocol in the **LAP entry link** and **GitHub link to LAP entry documents**

In our laboratory, this protocol has been used to perform plasmids using the Golden Standard modular cloning of levels 1 and level 2

The current version incorporates the following modifications:

- **Description of robot and protocol setup in a separate protocols.io entry** (Setting and Customizing OT-2 for LAP Entries)
- **New variables added**: *Position Distribute Water, Touch Tip After Distributing Water, Change Tip in Water Distribution, Position Distribute Reaction Mix, Touch Tip After Distributing Reaction Mix, Change Tip in Mix Distribution, Position Distribute Acceptor/Module, Touch Tip After Distributing Acceptor/Module, Change Tip in Acceptor/Module Distribution, Max Volume Per Mix Tube In Shaker*
- **Iteration through all the DNA parts, first distributing acceptors and then modules, to all the final combinations that include those parts**.
- **Tips are being replaced every time volume is aspirated from the second reagent onwards when transferred to the mix tubes, avoiding contamination of the original reagents.**

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)

This protocol has been tested by assembling level 1 and level 2 constructs with parts of the golden standard database

Materials

Software

- Python 3.7.1
- opentrons software version 7.0.2
- python packages: pandas (0.25.3), numpy (1.15.1), openpyxl (3.1.2), 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

- PCR skirted plate

Equipment

PCR 96-plate low profile Thermo Scientific	NAME
PCR Plate, 96-well, low profile	TYPE
Thermo Scientific	BRAND
AB0800R	SKU
https://www.thermofisher.com/order/catalog/product/AB0800R?SID=srch-srp-AB0800R	LINK

- Opentrons Eppendord Tube Rack

Equipment

Opentrons 24 Tube Rack with Eppendorf 1.5 mL Safe-Lock Snapcap	NAME
Tube Rack	TYPE
Opentrons	BRAND
opentrons_24_tuberack_eppendorf_1.5ml_safelock_sna	SKU
https://labware.opentrons.com/opentrons_24_tuberack_eppendorf_1.5ml_safelock_snapcap?category=tubeRack	LINK

- Tube Rack Eppendorf for Heater-Shaker + 24 eppendorf tube holder

TubeHolder_Eppendorfs_HS_OT2.stl

Equipment

24 Eppendorf Tube holder	NAME
Tube holder	TYPE
Opentrons	BRAND
999-00030	SKU
https://shop.opentrons.com/4-in-1-tube-rack-set/	LINK

- 1.5mL eppendorfs (without the cap) + 4ºC cold-block with adaptor (file attach)

Equipment

BRAND™ Centrifuge Tube Mini-Cooler	NAME
ColdBlock	TYPE
BRAND	BRAND
10141921	SKU
https://www.fishersci.es/shop/products/brandtech-scientific-brand-centrifuge-tube-mini-cooler-3/10141921	LINK

adaptor_OT_coldblock.stl

Reactives:

- **Water:** MilliQ water

- **T4 Ligase and Ligase Buffer:** T4 DNA Ligase, 100u **Promega Catalog #**M1801
- **Restriction Enzyme:** BpiI (BbsI) (10 U/µL) **Thermo Fisher Catalog #**ER1012
  BsaI-HFv2 **New England Biolabs Catalog #** R3733S

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

Equipment

Opentrons Thermocycler Module	NAME
Thermocycler	TYPE
Opentons	BRAND
999-00174	SKU
https://shop.opentrons.com/thermocycler-module-1/	LINK

Equipment

Opentrons Heater-Shaker Module	NAME
Heater-Shaker	TYPE
Opentrons	BRAND
999-00157	SKU
https://shop.opentrons.com/heater-shaker-module/	LINK

Safety warnings

If you are using the heater-shaker take in account that there is a limit of RPM that it can shake before the liquid of the eppendorfs get out. As well, take in account that there are constrictions that could prevent some labware to be placed.
This speed depends on the liquid consistency and volume

Before start

Being a one-pot reaction in this protocol only 1 restriction enzyme is used so one type of level, in the case of the golden standard database, can be produced in only 1 run

Files Preparation

1 **Preparing Customized Template**

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

Here there is attached 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. **PerPlateVariables:** variables related to the specifications of each source plate
3. **PipetteVariables:** variables related to the pipettes that are going to be used
4. **ReactionVariables:** variables that will determine the final mix of the wells
5. **ModuleVariables:** variables related to the modules used in the protocol, the thermocycler and the heater-shaker
6. **Combinations:** set of combinations that are going to be created in the final wells, one combination per row and one DNA part per cell
7. **Map DNA Parts Sheet(s):** sheet(s) with the names of each DNA part that can be used to create final assemblies denoted in the combinations sheet. The sheet(s) need to have also dthe name of the rows and columns of the plate and the wells that does not have any sample need to be left empty → *not included in the template but needed to be included and have the same names as established in the variable* ***Name Map DNA Parts*** *from the PerPlateVariables Sheet*
8. **TemperatureProfile (Optional):** a profile that will be performed in the thermocycler if set as True in the ModuleVariables sheet

TemplateVariablesMoCloAssembly....

MoCloAssemblyInstructions_v200....

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 *Store it with the name VariablesMoCloAssembly.xlsx*

Note

The file should be spelt **exactly** *VariablesMoCloAssembly.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 *VariablesMoCloAssembly.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

Biocomputation Lab

Preview

Running Protocol

3 **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 has been tested until version 7.0.2

Indications may vary from version to version

Software

NAME	OS	DEVELOPER	REPOSITORY
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-MoCloAssembly-OT2** was the 2.0.0 which script you can find attached

ScriptMoCloConstructAssembly_v2...

Note

The last script version can be found at the **Github entries directory** or **web repository section** of the LAP repository.
The name of the directory or entry should be **LAP-MoCloAssembly-OT2** followed by the version.

Software

NAME	DEVELOPER	REPOSITORY
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 *VariablesMoCloAssembly.xlsx* is → Proceed To Setup

After clicking on Proceed to Setup, you should obtain, if there is no error, the positions of the labware 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 be raised by the app. Many errors are contemplated already and have a specific message that will give the user a hint of what could have gone wrong.

Note

Both the volume of the reagents and the DNA parts are 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 and tube racks 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 needed)*

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 **9 parts**:

1. (Optional) The block temperature from the thermocycler reaches a temperature set by the user in case that variable is not empty
2. Distribute the needed water to each final well. The volume of water is calculated by the OT-2 according to the volume set in the variable *Volume Final Each Reaction (uL)* taking in account the volume of the other reagents and the number of DNA parts that is going to that specific well.

3. Creation of mix(es) transferring ligases, buffer, serum and restriction enzyme to new tube(s)
4. Mixing with either a pipette or heater-shaker
5. Distribute mix to final plate(s)
6. Distribute DNA parts to the corresponding wells
7. Generate identity maps to be exported
8. (Optional) Pause the program for user to put lids on the plate located in the thermocycler (if set like that in the input variable)
9. (Optional) Temperature profile with thermocycler module
10. (Optional) Block of thermocycler stay to the set temperature

Expected result

One or more plates where there is a mix between the different DNA parts and the mix of reagents needed to perform the MoClo assembly process, each combination will be in one well.

A **sheet for every final plate will be created as well in an Excel file** with the given name in the sheet GeneralVariables in the variable "Name File Final Constructs" followed by the extension .xlsx in the folder */data/user_storage* of the robot where we run the script.

After-Running

5 **Retrieve labware from the OT**

6 **Importing map from robot**

There will be a map with the name set in the variable *Name File Final Constructs* in the sheet *GeneralVariables* followed by the extension .xlsx: [*Name File Final Constructs*].*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 into account that files overwrite, so if you give the final map the same name in two consecutive runs, you will get only the results of the last run.

Example

6h 6m 10s

7 **We want to make 47 constructs that are all level 1 from 38 parts, some created in the lab, some from the golden standard database.**

Citation

Blázquez B, León DS, Torres-Bacete J, Gómez-Luengo Á, Kniewel R, Martínez I, Sordon S, Wilczak A, Salgado S, Huszcza E, Popłoński J, Prieto A, Nogales J (2023) . Golden Standard: a complete standard, portable, and interoperative MoClo tool for model and non-model proteobacteria.

https://doi.org/10.1093/nar/gkad758 LINK

We are going to use the heater-shaker module to mix the assembly mix (enzyme, ligase, serum and buffer) and use a thermocycler module to perform the construct assembly with a provided temperature program.

7.1 **Prepare variable file** 10m

Excel temple filled and saved with the name *VariablesMoCloAssembly.xlsx*

VariablesMoCloAssembly.xlsx

7.2 **Upload custom labware to app** 4m

We are using 2 custom labware called *3dprinted_opentrons_shaker_1.5mleppendorf* and *coolblocknew_12_tuberack_1500ul* that have been created with the labware creator that opentrons offers (**https://labware.opentrons.com/create/**)

3dprinted_opentrons_shaker_1.5mle...

coolblocknew_12_tuberack_1500ul.j...

We upload it to the opentrons app (make sure that it is in the robot app) and the robot system as stated in the protocol in step

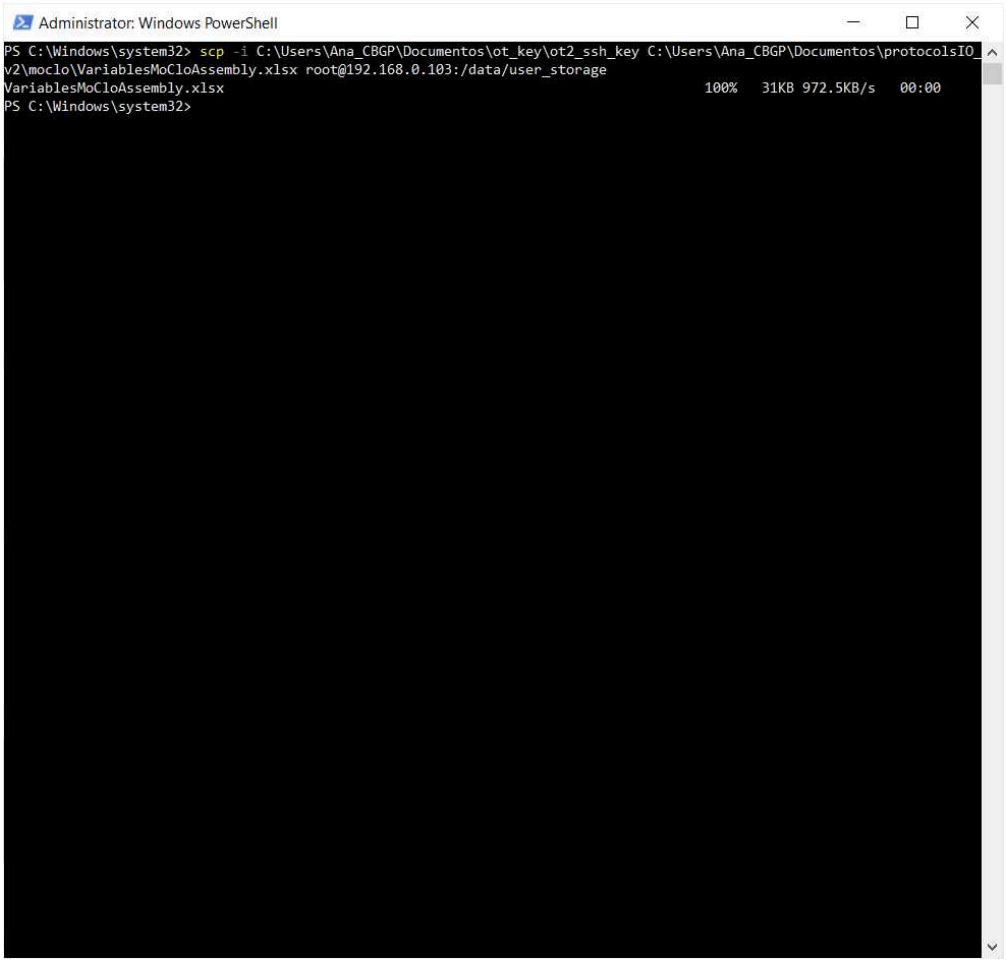
go to step #2 Setting and Customizing OT-2 for LAP Entries

3dprinted_opentrons_shaker_1.5mle...

coolblocknew_12_tuberack_1500ul....

(these are zip files because they 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)

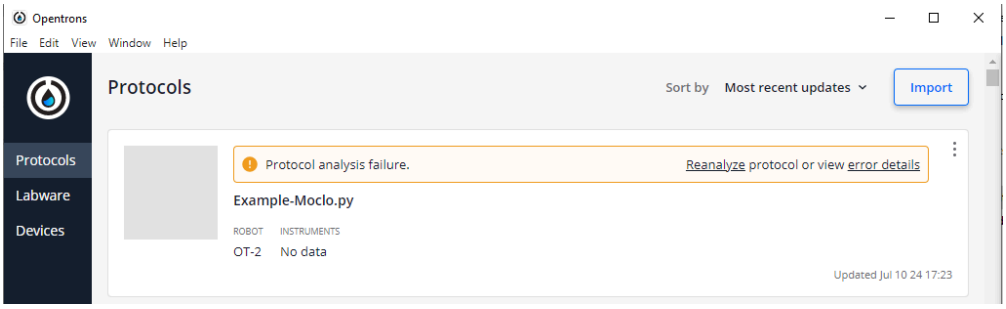
7.3 **Export the variable file to *the /data/user_storage* folder in the robot** 2m



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

7.4 **Import the script** we downloaded from go to step #3 (I named it *Example-Moclo.py*) to the OT-App. 1m

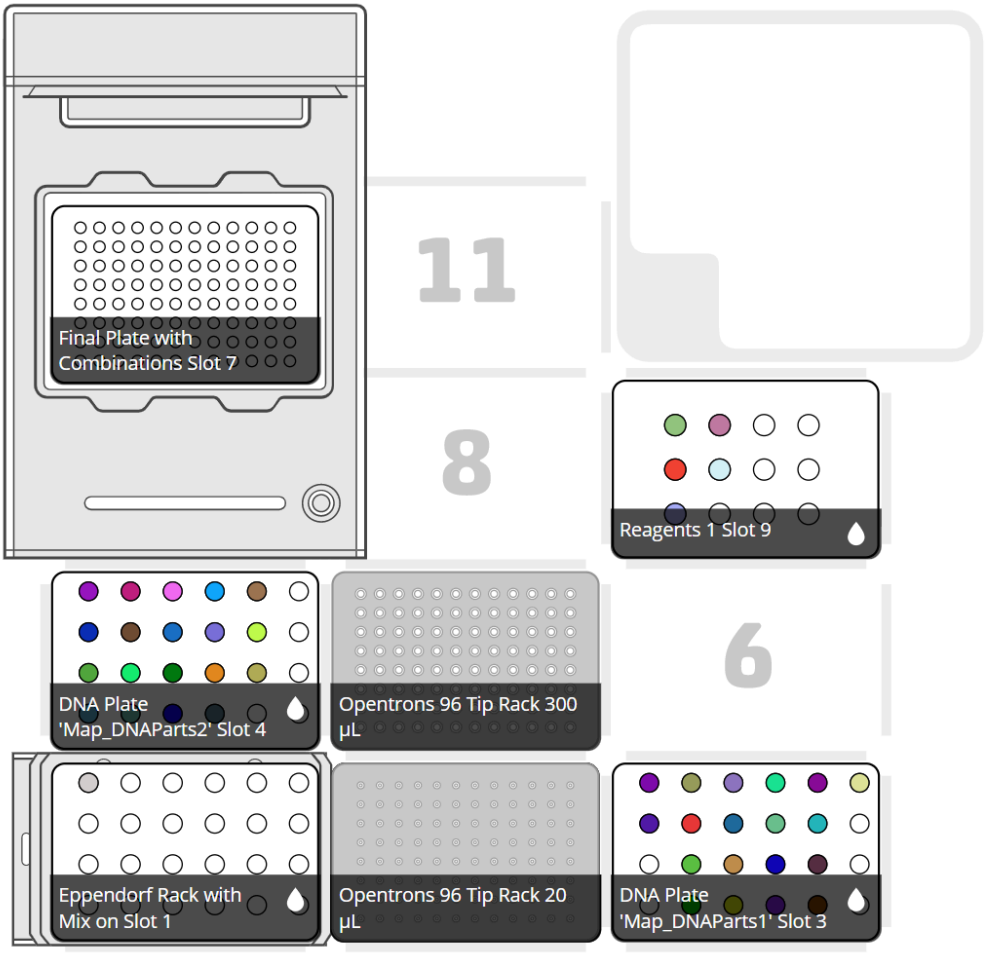
Example-Moclo.py



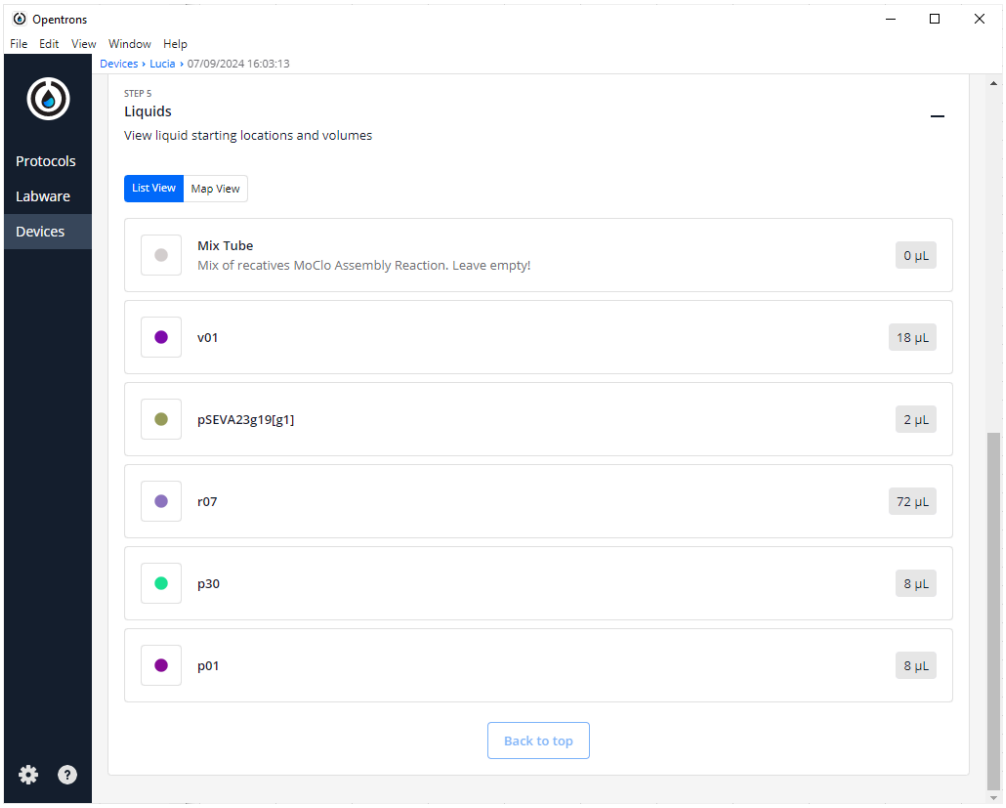
Result of importing the Python script in the OT-App

7.5 **Run the protocol** in the robot that we have transferred the Excel file 1m

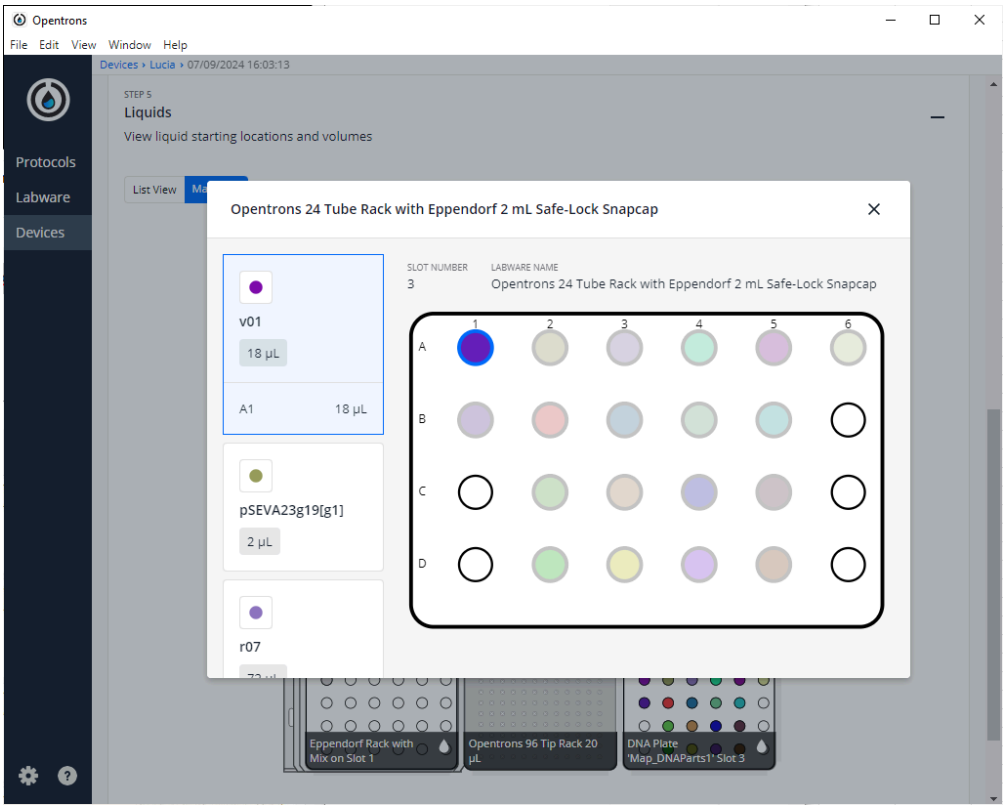
,



Labware and liquid set-up layout that corresponds to the run of the example variable input file



Volumes of the different reagents needed to perform the protocol that corresponds to the run of the example variable input file



Positions and volumes of different DNA parts. The volumes are calculated but the positions are the ones set in the input file

7.6 **Clean the platform** of the robot that we are going to perform the protocol 5m

7.7 **Prepare all reagents and labware** in the places as the App is showing, taking into account the notes in step go to step #3.2 Notes 7m

7.8 **Start Run** 20s

> Note
>
> The provided times from this point on are specific to the example, they can variate depending on the ammount of combinations, the variables related to the tip changing, the presence of a thermocycler, etc.

7.9 **Distributing water** 5m 30s

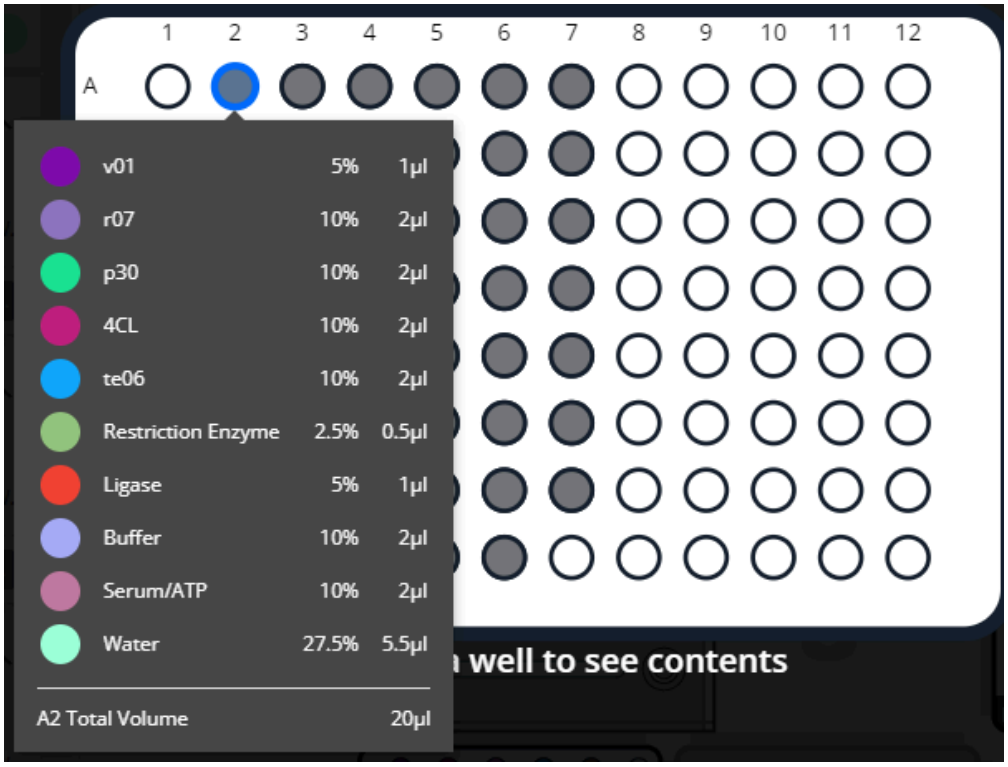
7.10 Creation and distribution of mix

7m 30s

7.11 Distribution of DNA parts (acceptor and modules)

1h 28m

Expected result



Example of the content of a well, in this case A1 in the labware *Final Plate 1 with Combinations Slot 7*

Here, we will obtain the modular cloning mix and the diffreent DNA parts that are set in the input file. 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 plate on slot 7 in this case

7.12 Temperature Profile

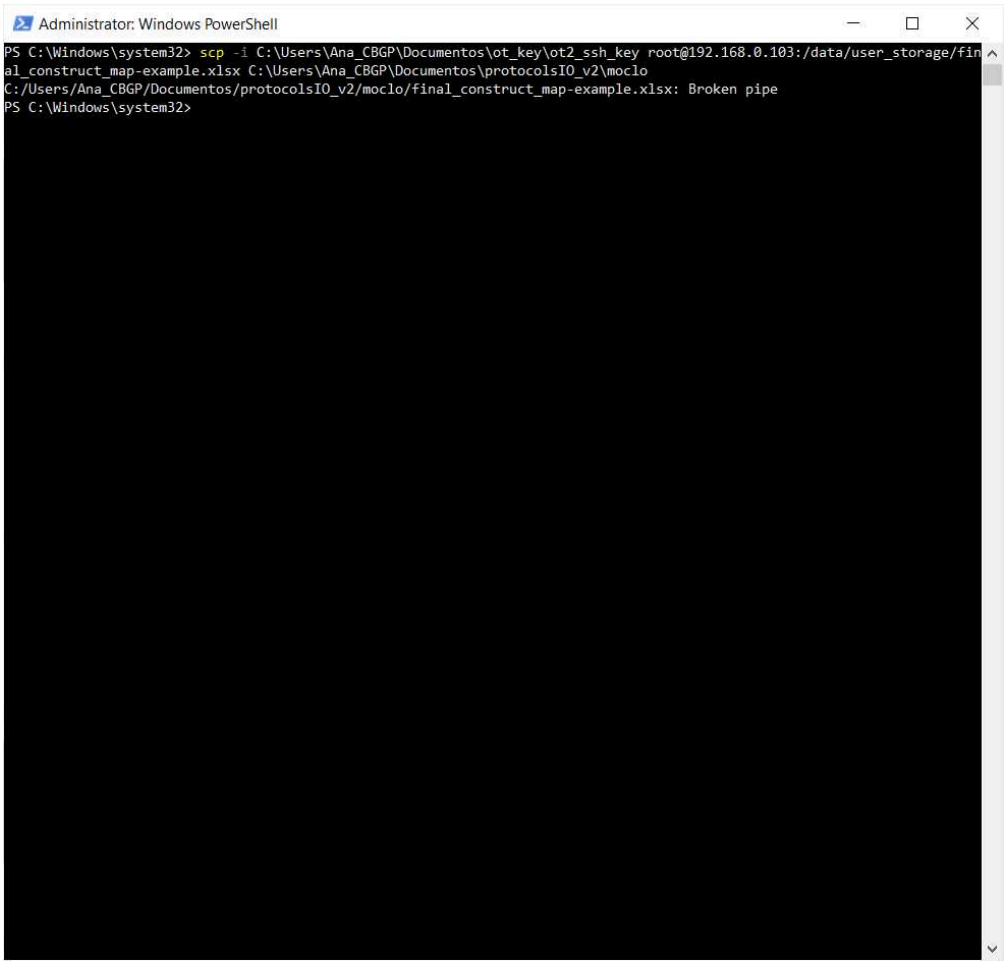
3h 50m

7.13 **Retrieve labwares** from the OT

5m

7.14 **Retrieve the final map(s) file from the robot** where we run the protocol. In this case, they will be called *final_construct_map-example.xlsx* (name that is stated in the variable file in the variable *Name File Final Constructs* located in the *GeneralVariablesSheet*)

1m



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

final_construct_map-example.xlsx

Protocol references

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

Golden Standard: a complete standard, portable, and interoperative MoClo tool for model and non-model proteobacteria *(Nucleic Acids Research)* https://doi.org/10.1093/nar/gkad758