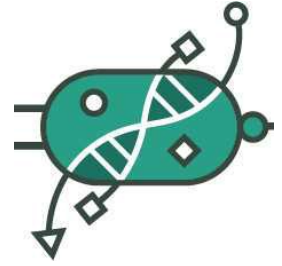


Aug 06, 2024

OT-2 Custom Mixing with Single or Multi-Channel Pipette

DOI

dx.doi.org/10.17504/protocols.io.5jyl82686l2w/v1



Ángel Goñi-Moreno¹, Ana Mariya Anhel², Paula Mugica-Galan²

¹CNB CSIC; ²CBGP UPM-INIA

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



biocomp.cbpg Biocomputation Lab

Centro de Biotecnología y Genómica de Plantas

OPEN  ACCESS



DOI: dx.doi.org/10.17504/protocols.io.5jyl82686l2w/v1

Protocol Citation: Ángel Goñi-Moreno, Ana Mariya Anhel, Paula Mugica-Galan 2024. OT-2 Custom Mixing with Single or Multi-Channel Pipette. [protocols.io https://dx.doi.org/10.17504/protocols.io.5jyl82686l2w/v1](https://dx.doi.org/10.17504/protocols.io.5jyl82686l2w/v1)

License: This is an open access protocol distributed under the terms of the **[Creative Commons Attribution License](#)**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: July 25, 2024

Last Modified: August 06, 2024

Protocol Integer ID: 104076

**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 created to help efficiently produce high-throughput plates with different reagents using either a single or multi-channel pipette controlled by an Opentrons OT-2 robot. By running this script, users can achieve a precise layout of reagents in specified volumes across the final plate(s) using a single-channel or multi-channel pipette based on a customizable Excel input file.

The protocol uses a Python script for the Opentrons OT-2 robot in conjunction with an Excel file that contains all the necessary parameters and will be processed by the Python script.

This Excel file contains customizable variables such as the number of final plates, transfer volumes, plate types, and the type of pipette used.

This method ensures the accurate distribution of reagents based on user-defined configurations, making the high-throughput process more efficient.

This protocol has been used in our laboratory to check how different quorum sensing systems work in *P. putida* KT2440.

This protocol is a set of instructions or description of the [LAP repository](#) entry **LAP-CustomReagentMixingMultiSinglePip-OT2-1.0.0**

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](#)

Guidelines

This protocol was developed using Python 3.7.1, OT App Software Version 7.0.2, and API level version 2.14 on 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 testing quorum sensing systems with different concentrations of 3mBz to control the systems programmed in them.

The maximum number of 96-well plates per run is 9 final plates with no replicas, replacing the tiprack, using only a single-channel pipette and one falcon tube rack necessary to distribute the reagents to all the final plates.



Materials

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, itertools, time
- OT App version 7.0.2
- Excel

OT-2 Labware

- Opentrons Tip racks
- Opentrons Pipettes
- Opentrons Tube-racks
- Well plates (such as Corning 96 Well Plate 360 μ L Flat)
- Reservoirs (such as a NEST 12 Well Reservoir 15 mL)
- Opentrons HEPA filter (Optional)

Files Preparation

1 Preparing Customized Template

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

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

1. **GeneralVariables:** These are variables mainly related to the labware to be used, optimization for the source plate, and sterility variables.
2. **PipetteVariables:** These are variables related to the pipettes to be used.
3. **FinalPlatesVariables:** variables related to the specifications of each final plate to be created, such as the maps associated with this final plate, the number of replicas to make and the type of pipette used to create each.
4. **MapsReagents:** These are the sheets that are established in variables in FinalPlatesVariables that define the layout of reagents to transfer/distribute to each one of the final cells
5. **MapsVolumes:** These are the sheets established in variables in FinalPlatesVariables that define the layout of volumes of each reagent defined in MapReagents to transfer/distribute to each of the final cells.



TemplateVariablesCustomMixing.xlsx



CustomMixingInstructions_v100.pdf

Note

The most updated Excel template can be found on the [LAPrepo Repository Page](#)

1.1 *Fill the template with the corresponding values*

1.2 *Save it with the name VariablesCustomMixing.xlsx*

Note

The file should be spelt precisely as **VariablesCustomMixing.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 file *VariablesCustomMixing.xlsx* to the directory */data/user_storage* of the OT system we will use to perform the protocol.

Also, if we use custom labware, we must 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 must ensure the package ***openpyxl*** is installed in the robot system.

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

Protocol



NAME

Setting and Customizing OT-2 for LAP Entries

CREATED BY

biocomp.cbpg 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 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 script version.



Software

Opentrons App

NAME

Windows >=10, Mac >=10 , Ubuntu >=12.04

OS

Opentrons

DEVELOPER

<https://opentrons.com/ot-app/>

SOURCE LINK

3.1 *Load the script in the App*

Protocols -> Import -> Drag Python script

This protocol version was developed when the last version of **LAP-CustomReagentMixingMultiSinglePip-OT2** was 1.0.0; the script is attached.



ScriptMixingReagents_v100.py

The name of the Python file is the 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-CustomReagentMixingMultiSinglePip-OT2**, followed by the version.

Also, the latest version of the script can be found at <https://www.laprepo.com/repository/>, with the same name as on GitHub.



Software

LAP Repository

NAME

<https://biocomputationlab.com/>

DEVELOPER

www.laprepo.com

SOURCE LINK

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

VariablesCustomMixing.xlsx is -> Proceed To Setup to overpour to account for any pipetting errors slightly

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

If the protocol with the set variables cannot run, an error will occur during the simulation's run in the app after *Start setup*. Many errors are already contemplated and have a specific message that hints to the user what could have gone wrong.

Note

The reagent volumes in the reservoirs for the multi-channel pipettes or tubes for the single-channel pipettes are precisely measured, so it is **recommended to overpour slightly to account for any pipetting errors.**

**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 to prepare the experiment layout. 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 *Turn the HEPA Filter on*



4.4 *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.

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

4.6 *Start Run*

The procedure that the robot is going to do is mainly divided into 2 parts:

1. Distribute reagents with single-channel if any final plate is set to be created with it.
2. Distribute reagents with multi-channel if any final plate is set to be created with it.



Expected result

One or more plates, with the reactive-volume layout set by the user in the Excel input file.

After-running

5 Retrieve labware from the OT

Example

6 We want to create three plates: 1 with a single-channel pipette and 2 with a multi-channel pipette.

The two plates filled with the multi-channel will have different concentrations of IPTG and certain bacteria. At the same time, the single-channel pipette will create a plate with varying combinations of reagents.

The multi-channel reservoir source's optimization will be low, and the reservoir will have 15 columns and 1 row.

The tip will be changed every time a reagent is changed, and the liquids will be dispensed at the top so the source reagents are not contaminated.

We will use a computer with a Windows 10 system.

6.1 Prepare variable file

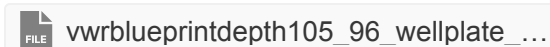
Excel template that we can find  [go to step #1](#) filled and saved with the name *VariablesCustomMixing.xlsx*



6.2 Upload custom labware to app

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/>)





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

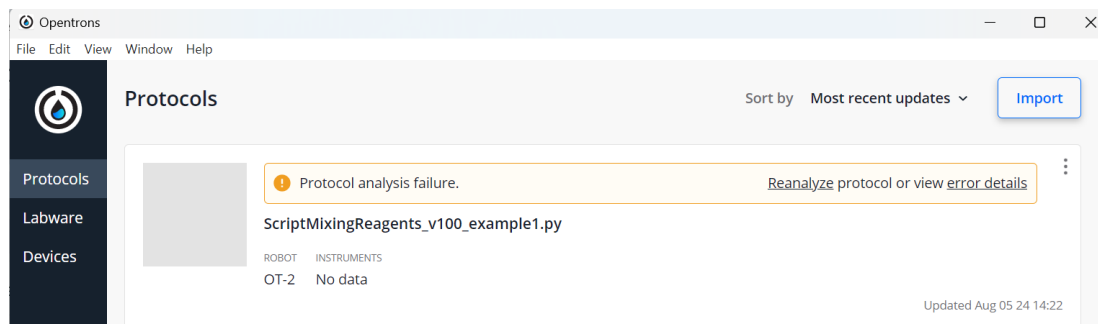
6.3 *Export the Excel file to the robot*

Export the *VariablesCustomMixing.xlsx* file to the */data/user_storage* directory in the robot where the script is going to be played following the instructions in

⇒ [go to step #2](#) Setting and Customizing OT-2 for LAP Entries

6.4 *Import the script to the robot*

 ScriptMixingReagents_v100_exempl...



Result of importing the Python script in the OT-App

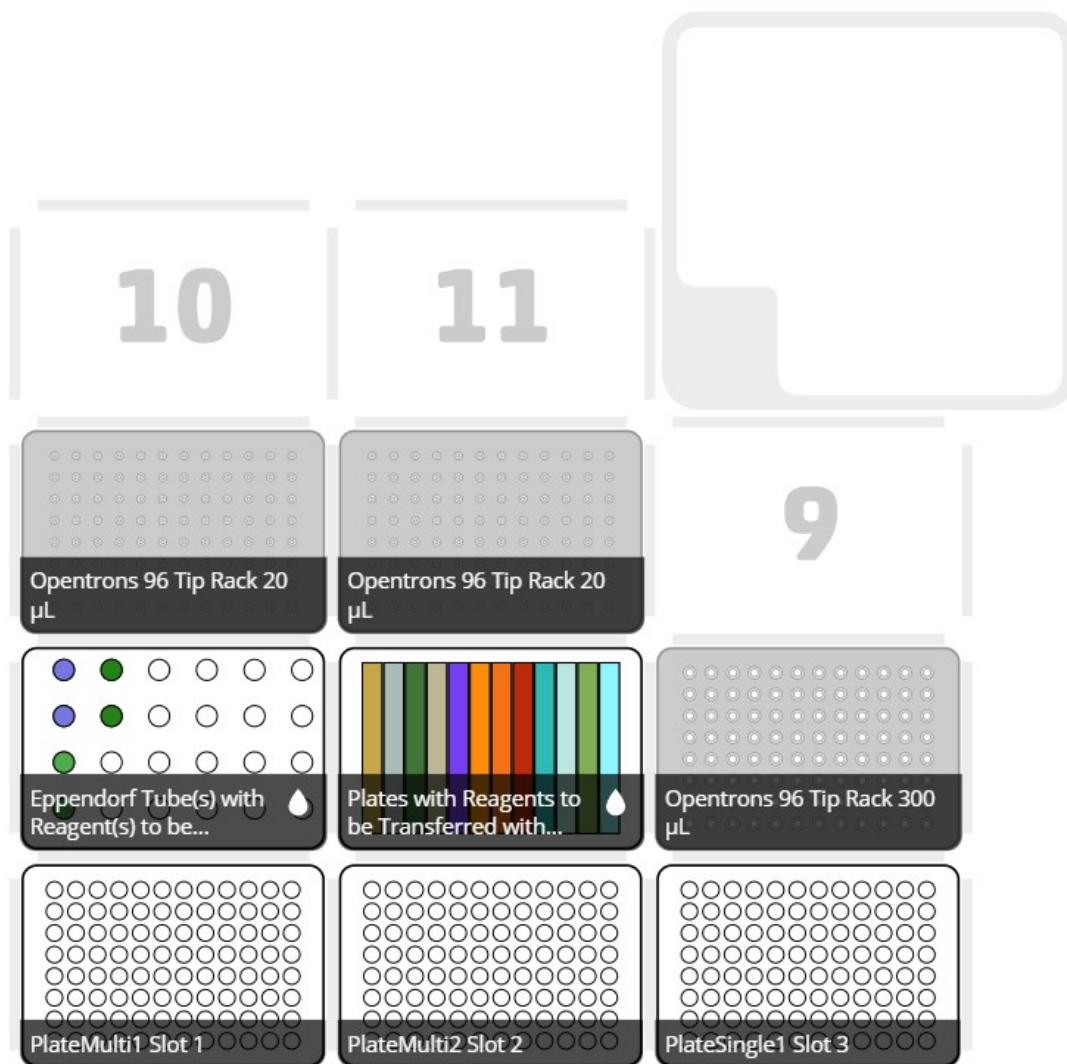
Note

An error during the simulation of the script in the app is normal. The script is meant to work in the opentrons robot, not on your computer.

6.5 *Run the protocol*

ScriptMixingReagents_v100_example1.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



reag_A

Medium reag_A

1590 μ L

A1

1415 μ L

B1

175 μ L

reag_B

Medium reag_B

1450 μ L

SLOT NUMBER

4

LABWARE NAME

Opentrons 24 Tube Rack with Eppendorf 1.5 mL Safe-Lock Snapcap

	1	2	3	4	5	6
A						
B						
C						
D						

Back to top

1590 μ L

reag_B

Medium reag_B

1450 μ L

reag_C

Medium reag_C

1480 μ L

A2

1380 μ L

B2

100 μ L

SLOT NUMBER

4

LABWARE NAME

Opentrons 24 Tube Rack with Eppendorf 1.5 mL Safe-Lock Snapcap

	1	2	3	4	5	6
A						
B						
C						
D						

Back to top



NEST 12 Well Reservoir 15 mL

SLOT NUMBER 5 LABWARE NAME NEST 12 Well Reservoir 15 mL

amp
Medium amp
2400 µL
A4 2400 µL

1mM
Medium 1mM
1200 µL

Slot Number	Labware Name
1	NEST 12 Well Reservoir 15 mL
2	NEST 12 Well Reservoir 15 mL
3	NEST 12 Well Reservoir 15 mL
4	NEST 12 Well Reservoir 15 mL
5	NEST 12 Well Reservoir 15 mL
6	NEST 12 Well Reservoir 15 mL
7	NEST 12 Well Reservoir 15 mL
8	NEST 12 Well Reservoir 15 mL
9	NEST 12 Well Reservoir 15 mL
10	NEST 12 Well Reservoir 15 mL
11	NEST 12 Well Reservoir 15 mL
12	NEST 12 Well Reservoir 15 mL

SLOT NUMBER

LABWARE NAME

5

NEST 12 Well Reservoir 15 mL

0.6mM
Medium 0.6mM
400 µL

A6400 µL

0.36mM
Medium 0.36mM
1200 µL

12 Wells

12

11

10

9

8

7

6

5

4

3

2

1

A



[illegible]



6.6 *Turn the HEPA filter on*





6.7 *Clean the deck*

6.8 *Perform calibrations if needed*



6.9 *Prepare all reagents and labware*

The places and volumes should correspond to the ones shown in the  pictures.

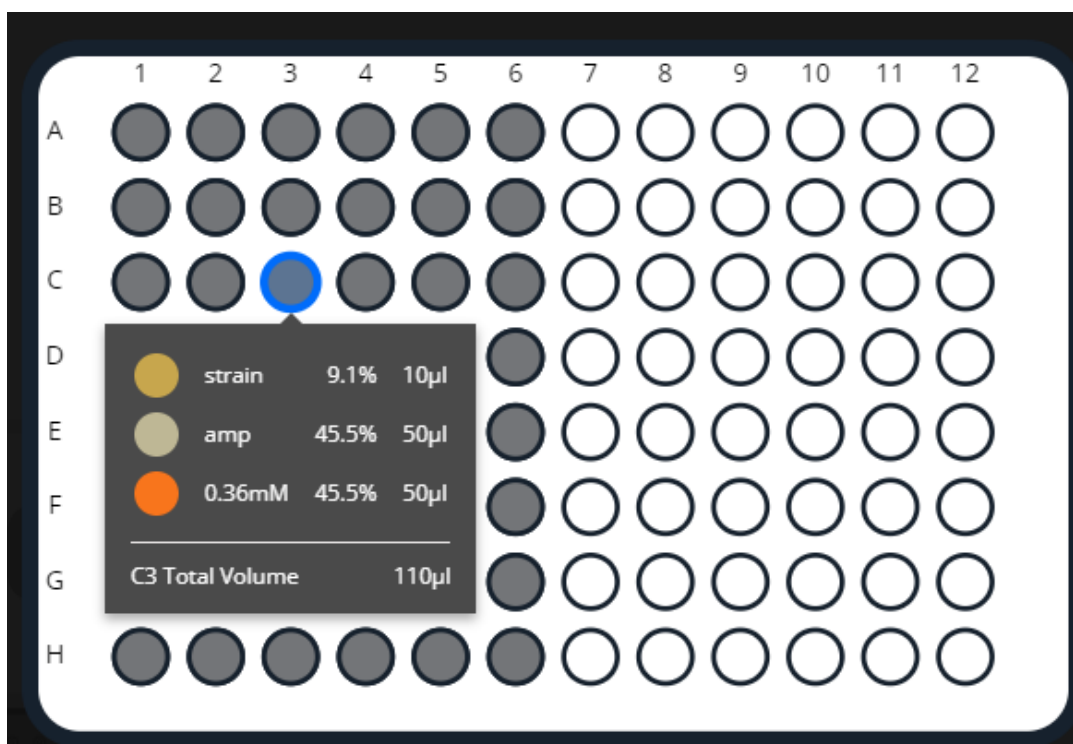
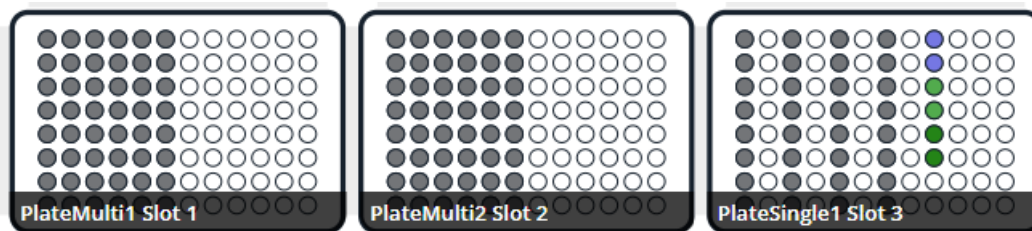
Place everything, taking into account the notes in step  [go to step #3.2 Notes](#)

6.10 *Start Run*

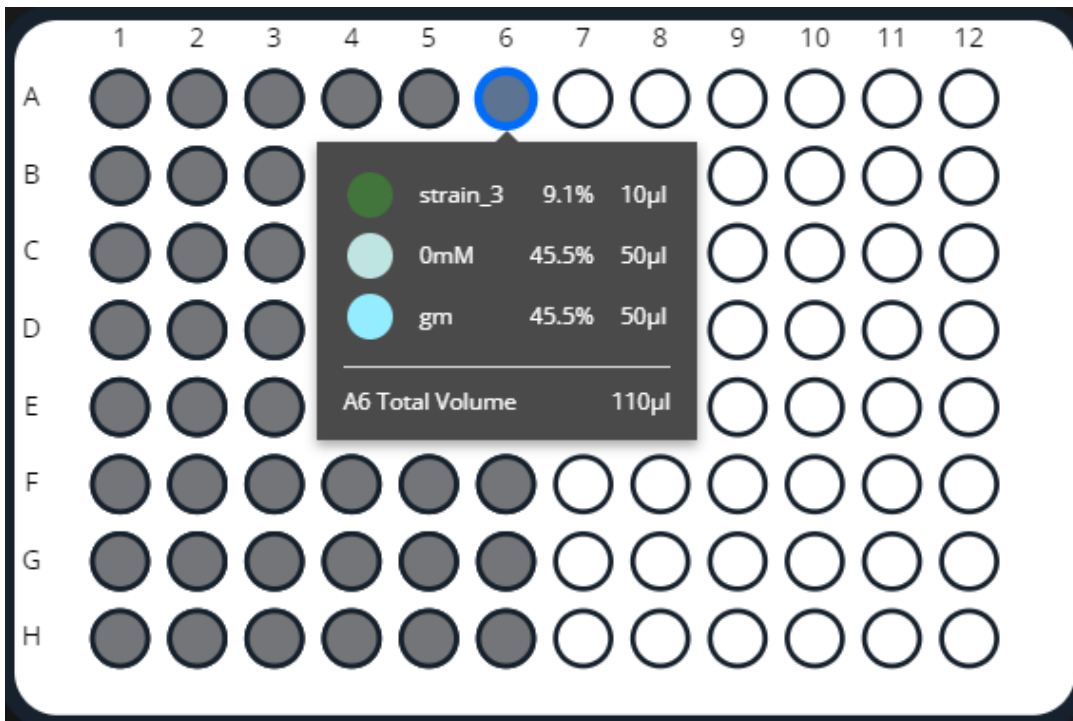
.

Expected result

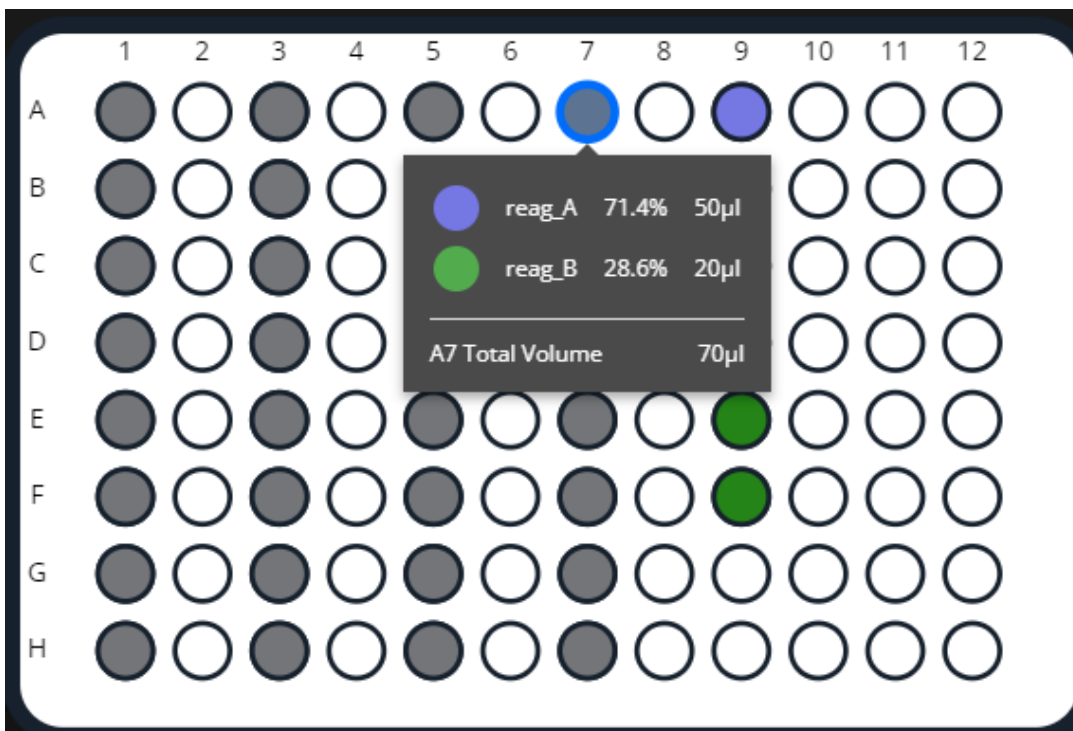
After the run has finished, we should obtain 3 final plates with the reagents and volumes described in the input Excel file.



Final content of C3 in PlateMulti1 located in Slot 1



Final content of A6 in PlateMulti2 located in Slot 2



Final content of A7 in PlateSingle1 located in Slot 3



6.11 *Retrieve Labware from the robot*

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>