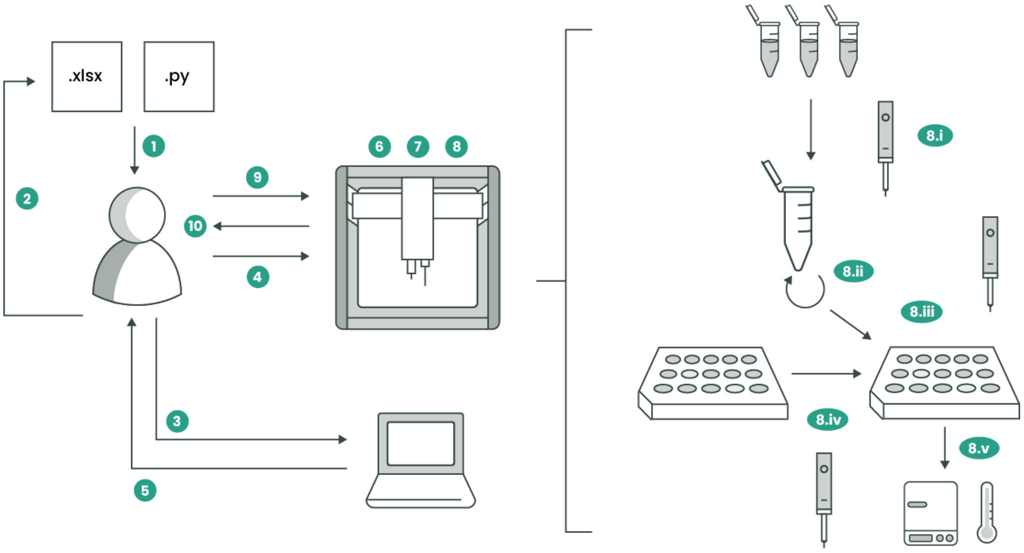
Instructions to use



1. Download the Excel template and Python script.

2. Fill the Excel file with the values of your variable to customise the experiment according to the guidelines at the [end of this document](#).

3. Import the Python script to the Opentrons App. A warning will appear on the protocol simulation because the script is not designed to read the Excel variable file from your computer but from the robot system.

4. Send the file to the robot that you are going to perform the protocol. If you are using a Unix system, you can send the file using the command `scp` to the folder `/data/user_storage`

   The command should look similar to:

   ```
   scp -i [password file] VariablesPCR.xlsx
   root@[IP_Robot]:/data/user_storage
   ```

   For more information about sending files to an Opentrons robot visit https://support.opentrons.com/s/article/Copying-files-to-and-from-your-OT-2-with-SCP

5. The Opentrons app will simulate the script in the chosen robot, providing you with an error message if something went wrong or with the required labware layout and needed reagents. The colours of the

regents can change due to them being generated randomly, but the volumes will always be the same given the same Excel file.
The volumes of the reagents are exactly what the run needs, so it is recommended to pour in the Eppendorf tubes between 20-50ul more of that reagent.

On the other hand, the liquid of the samples in the source plates corresponds to 90% of the max volume that the well can hold, it is only a recommendation, it does not mean that is the volume needed of that sample.

6. Set the different reagents and labware as the App shows. Both labware and reagents will have labels to help you recognise where every one of them should be placed.
7. Perform a labware offset, ensuring all the labware is calibrated correctly.
8. Close the door of the Opentrons and press the button 'Start run'. The Opentrons robot will follow the following stages:
   i) Transfer different reagents to create the primer sets, these sets can be either in the heater-shaker or in the same labware as the other reagents.
   ii) Mix primer sets, either with the heater-shaker or the pipette
   iii) Distribute the primer sets to the final labware. This labware can be placed in different slots in the Opentrons OT-2 or in the thermocycler.
   iv) Distribute to the primer set (s) the DNA templates.
   v) (Optional) Perform the temperature profile in the thermocycler.
9. Wait for the protocol to finish and, if needed, change the tip racks as stated in the App.
10. Import the maps of the combinations primer sets plus DNA templates in the final labware the same way you transferred the variable file to the Opentrons system. It should look like the following command:

```
scp -i [password file]
root@[IP_Robot]:/data/user_storage/[name_map].xlsx [user
computer path]
```

The name of the file is settled in the Excel variable file with an XLSX extension. This final EXCEL file will be composed with the maps of the samples and their sets on the different final plates.

The sheets will be named *FinalMapSlot* followed by the slot place that the plate was in the protocol run.

Guidelines to fill in the Excel file

General Variables Sheet

API Name Source Plate

- **Type of Value**: String
- **Description**: Opentrons API name of the labware with the samples that will be transferred and mix with the different set(s) of primer(s)
- **Example**: `vwrblueprintdepth105_96_wellplate_390ul`

Number of Source Plates

- **Type of Value**: Number (Integer)
- **Description**: The total number of source plates that will contain the DNA templates that will be combined with the PCR mix.
- **Example**: `2`

API Name Final PCR Plate

- **Type of Value**: String
- **Description**: Opentrons API name of the final labware in which the selected samples of the source plates will be transferred in addition to the sets.
- **Example**: `biorad_96_wellplate_200ul_pcr`

Well Start Final PCR Plate

- **Type of Value**: String
- **Description**: First well in the final labware in which a sample and the PCR mix will be placed in the first final plate.
- **Example**: `A2`

API Name Eppendorf Reagents Rack

- **Type of Value**: String
- **Description**: Opentrons API Name that will contain the Eppendorf tubes that will have the reagents needed to create the PCR mix that will be distributed to the final labware.
  In case the heater-shaker is set as False, the tubes associated with the set(s) of primers will be in this labware as well.
- **Example**: `coolblocknew_12_tuberack_1500ul`

Final Map Name

- **Type of Value**: String
- **Description**: Name of the final map, without the extension because it will always be an EXCEL file, in which we will have the names and locations of the DNA templates with the corresponding primer set. The final excel file will contain 1 Sheet per final plate.
- **Example**: `finalMapPCR`

Samples Plate Variables Sheet

Number Samples

- **Type of Value**: Number (Integer)
- **Description**: Number of samples that the plate contains, including control samples and the ones that are not going to be picked to perform a sample.
  Take in account that this number will be considered from the well stated in "Well Start" and will not consider that there is a well empty in between. For instance, if this value is set as "54" and the variable "Well Start" is set as A2, the protocol will consider that there are 54 samples in consecutive wells starting from the well A2, this one being included. The number of samples are counted top to bottom, left to right.
- **Example**: `54`

Well Start

- **Type of Value**: String
- **Description**: First well from this source plate that contains a sample that should be used as a DNA template in the final labware.
- **Example**: `A2`

Position Controls

- **Type of Value**: String | List
- **Description**: Position(s) in the source plate that contain control(s) of the protocol. They will be dispensed at the end of every set of primers in the final plate(s).
- **Example**: `G12`

Wells not to be perform PCR

- **Type of Value**: String | List

- **Description**: Position(s) in the source plate that will not be picked to perform PCR.
- **Example**: `A2, B3`

Map IDs

- **Type of Value**: String
- **Description**: Name of the sheet that will hold the names or IDs of the samples that will be annotated in the final map(s).
  This is an optional variable, if not provided, the name of the sample will be the name of the well with the place that it has in the layout of the run, for instance, "A2 Slot 1".
  This map (s) needs to have the same number of rows and columns as the labware set in the variable "API Name Source Plate" and should have the names of those rows and columns.
- **Example**: `mapNamesPlate1`

* If needed, this sheet can have more columns to the right, this meaning that if the Opentrons have more than 11 available slots, the sheet can be expanded and filled. This column adding does not guarantee that more final plates can be created because that depends on other variables, such as the tip rack replacement value or the presence of thermocycler.
There is a need of as much additional sheets with the set names as set in the variable 'Map IDs'.
The columns of the plate can have any name, these names will be included in the final layout that you will be able to see in the OT-App to help you recognize where each source plate must be placed. Nevertheless, if there is an error that is cause by any of the variables of this sheet, the name of the source plate will be given by the index of the column, being Source Plate 1 the one in furthest on the left.

Pipette Variables Sheet

API Name Right Pipette

- **Type of Value**: String
- **Description**: Opentrons API name of the pipette that will be in the right mount of the Opentrons arm.
  If this variable is left empty both 'API Name Tiprack Right Pipette' and 'Initial Tip Right Pipette' will be ignored.
- **Example**: `p300_single_gen2`

API Name Tiprack Right Pipette

- **Type of Value**: String
- **Description**: Opentrons API name of the tip rack associated with the right mount pipette.
  This script does not check that the pipette and the tip rack are compatible. In case of pipette and tip rack incompatibility errors like "Arc out of bounds in the Z-axis" could happen.
  If the tip rack from the right and left pipette are the same, both pipettes will pick up from the same tip rack, and the initial tip should be the same.
- **Example**: `opentrons_96_tiprack_300ul`

Initial Tip Right Pipette

- **Type of Value**: String
- **Description**: First tip that will be picked by the right pipette.
- **Example**: `A1`

API Name Left Pipette

- **Type of Value**: String
- **Description**: Opentrons API name of the pipette that will be in the left mount of the Opentrons arm.
  If this variable is left empty both 'API Name Tiprack Left Pipette' and 'Initial Tip Left Pipette' will be ignored.
- **Example**: `p20_single_gen2`

API Name Tiprack Left Pipette

- **Type of Value**: String
- **Description**: Opentrons name of the tip rack from which the left pipette will pick up tips from.
  This script does not check that the pipette and the tip rack are compatible. In case of pipette and tip rack incompatibility errors like "Arc out of bounds in the Z-axis" could happen.

If the tip rack from the right and left pipette are the same, both pipettes will pick up from the same tip rack, and the initial tip should be the same.

- **Example**: `opentrons_96_tiprack_20ul`

Initial Tip Left Pipette

- **Type of Value**: String
- **Description**: First tip that will be picked by the left pipette.
- **Example**: `A1`

Replace Tiprack

- **Type of Value**: Boolean
- **Description**: Value that will determine that, in case of more than one tip rack needed for the run of the protocol, the tip rack will be replaced and not added to the labware layout. If the right and left pipette tip racks are the same, and this variable is set to True, only one slot in the layout will be occupied by tips.
  This variable only accepts two values, True (tip racks will be replaced) or False (tip racks will be added to the layout).
- **Example**: `True`

Reagents Per Reaction Sheet

Number primer/set

- **Type of Value**: Number (Integer)
- **Description**: The number of primers that each set must have.
- **Example**: `2`

Number sets

- **Type of Value**: Number (Integer)
- **Description**: The number of different sets that the DNA templates are going to be mix with in the protocol.
- **Example**: `1`

Volume each primer (uL)

- **Type of Value:** Number

- **Description:** Volume, in uL, that the final volume in each well will have of each primer
- **Example**: `2`

Volume polymerase mix (uL)

- **Type of Value**: Number
- **Description**: Volume, in uL, of the amount of polymerase mix that each final well will have. This reactive is going to be transferred with a lower aspiration and dispense flow rate because of its usually high viscosity.
- **Example**: `10`

Volume sample DNA Template (uL)

- **Type of Value**: Number
- **Description**: Volume, in uL, of the DNA template that will be transferred to each primer set in the final labware(s).
- **Example**: `2`

Final Volume (uL)

- **Type of Value**: Number
- **Description**: Final volume, in uL, that each final well will have at the end of the run, this volume includes the DNA template and the volume of mix (polymerase + primers + water if needed). Each well will be filled with water until it reaches this volume.
- **Example**: `20`

Extra Pipetting Factor

- **Type of Value**: Number
- **Description**: Fraction of volume added to the total of volume needed for the reaction volume destined to make sure the pipetting error does not affect in the different volumes.
  The range of acceptable values is [0, 1)
- **Example**: `0.2`

Module Variables Sheet

Presence Thermocycler

- **Type of Value**: Boolean
- **Definition:** We are defining if the run is going to use the thermocycler module of the OT or not.

Only two values are accepted in this variable: True or False.
In case of True, the run will only have one final plate. In addition, another Sheet in Excel called “TemperatureProfile” should be filled.
In case of False, that sheet will not be read, so it can be deleted or left empty and more than 1 final plates can be obtained. In addition, the values of the variables ‘Final Open Lid’, ‘Temperature Lid’, ‘Hold Block Temperature’ and ‘Pause Before Temperature Program’ will be ignored and can be left empty.
Take in account that this module occupies 4 slots and only 1 final plate can be created. In addition, there are other kind of restrictions that are associated with having this module present, such as not being able to put the heater-shaker module in the slot 4 because of dimension conflicts.
For more information about other conflicts that can arise with the thermocycler module check the Opentrons guide.

- **Example**: `True`

Presence Heater-Shaker

- **Type of Value**: Boolean
- **Definition**: Value that will define if the primer set (s) mix will be performed by a heater shaker, variable set as True, or by a pipette, variable set as False.
  Only two values are accepted in this variable: True or False.
  If the variable is set as False, the tubes that corresponds to the set(s) will be placed in the same labware as the rest of the reagents (polymerase, water, and primers)
  In case this variable is set as False is important than the labware set in ‘API Name Eppendorf Reagents Rack’ contains 1.5mL eppendorfs because the pipette is optimized only to mix these tubes.
  In addition, the variables ‘RPM Heater-Shaker’ and ‘API Name Heater-Shaker Labware’ will be ignored and can be left empty if this variable is set as False
  Take in account that setting this variable as True come with layout restrictions, such as some type of labware not being able to be placed next to this module.
  For more information about other conflicts that can arise with this module check the Opentrons guide.
- **Example**: `True`

Final Open Lid

- **Type of Value**: Boolean

- **Definition**: Value that will define if the lid should be open (True) or closed (False) at the end of the temperature profile run by the thermocycler. Only three values are accepted: True, False, or empty cell. The latter, in case "Presence of Thermocycler" is set as True will by default set as False.
- **Example**: `False`

Temperature Lid

- **Type of Value**: Number
- **Definition**: Temperature that the lid of the thermocycler will set when performing the temperature profile if there is a thermocycler. Make sure that the thermocycler can reach that temperature.
- **Example**: `100`

Hold Block Temperature

- **Type of Value**: Number
- **Definition**: Final temperature of the block that is going to be set at the end of the profile. This temperature will be maintained until the block is deactivated by the user, even when the run finishes.
  If empty, the lid will be deactivated after the run, making the final temperature will be environment temperature.
- **Example**: `4`

RPM Heater-Shaker

- **Type of Value**: Number
- **Definition**: Value of the RPM that the Heater-Shaker will establish to shake the reagents mix for 15 seconds every time the program needs to take volume from a new final mix eppendorf tube. Make sure that the heater-shaker can reach this RPM and that the volume of mix in the tube(s) does not spills out during the mixing.
- **Example**: `1200`

API Name Heater-Shaker Labware

- **Type of Value**: String
- **Definition**: Opentrons API Name of the Labware where the primer sets will be set, mix, and distributed from. Make sure it is compatible with the heater-shaker module.
- **Example**: `opentrons_24_tuberack_1500ul_shaker`

Pause Before Temperature Program

- **Type of Value**: Boolean
- **Definition**: This pause is intended for mixing of the PCR sets with the DNA template (the protocol does not do it) and/or putting caps to the labware for less evaporation.
  Only two values are accepted for this variable: True or False.
  If Thermocycler Presence is set as False, this value can be left empty.
- **Example**: `True`

Temperature Profile Sheet

Temperature

- **Type of Value**: Number
- **Definition**: Temperature that the block in the thermocycler will reach in this step of the program.
- **Example**: `98`

Time (s)

- **Type of Value**: Number (Integer)
- **Definition**: Time, in seconds, that the temperature of this step will be held.
- **Example**: `90`

Number of Cycles

- **Type of Value**: Number | Character
- **Definition**: How many cycles the set of steps will be performed. This variable should only be established when "Cycle Status" is set as "End". In all of the other cases should be filled with the character "-"
- **Example**: `6`

Cycle Status

- **Type of Value**: Character
- **Definition**: This variable will define when a cycle starts and ends. When we want to start a cycle that will contain one or more steps, we should be put in this variable "Start".
  When we want it to end, meaning that the current step will be the last one of the cycles we need to set this variable as "End".
  In all the other cases, we need to have the cell with the character "-".

All the steps between a "Start" and "End" row/step will be repeated as many times as stated in the variable "Number of Cycles" when this value is set as "End".

- **Example**: `End`

*This sheet is optional and is only going to be read when the variable "Presence of Thermocycler" in the sheet ModuleVariables is set as True. In this case, the sheet name should be strictly "TemperatureProfile". Otherwise, this sheet can be erased from the Excel file.

Map ID Sheet(s)

Sheet with the name set in the variable "Map IDs" in the Sheet "SamplesPlateVariables".

This sheet should have the same dimensions (rows and columns) as the labware set in the variable "API Name Source Plate" in the sheet "GeneralVariables".

In addition, it should have the names of the columns and rows of that labware as it is defined in the Opentrons labware definition. For example, a 96 well plate can have the columns named 1 to 12 and the rows named A to H.

If a well it is left empty in a map and the sample is transferred to the final plate, the ID will be the name of the well, for instance, A1.

If 2 samples have the same name in the same source plate you will not be able to differentiate those samples. However, if 2 samples have the same name on different source plates, that problem will not occur because the ID of the sample is followed by the slot from which the sample has been taken.