Massively Parallel Qubit DNA Quantification Using OpenTrons Version 4

Suzy Huang

Abstract

Measurement of DNA can be done using two broad methods (Spectrophotometric or Fluorometric).

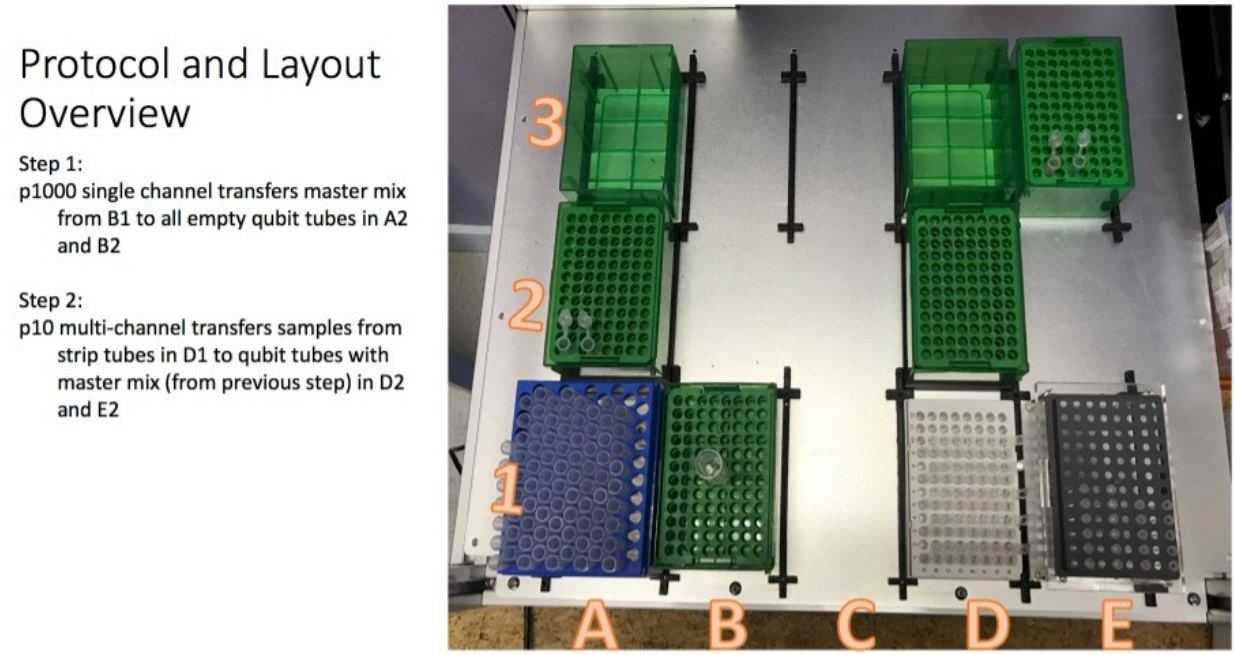
Qubit is a fluorometric method and uses a double stranded DNA specific dye. Therefore the sensitivity is higher and impurities like RNA or nucleotides in the sample will not affect the quantification as it would in spectrophotometric methods like a Nanodrop.

The Qubit is first calibrated using two standards. To make a measurement, 199ul of buffer +1ul of dye + 1ul of sample is added to a qubit tube, which is inserted into the instrument.

Here, we use OpenTrons to prepare samples for qubit measurement. It also saves time by requiring only one initial calibration for each container. It uses p1000 to distribute master mix and p10 to distribute unique samples.

Citation: Suzy Huang Massively Parallel Qubit DNA Quantification Using OpenTrons. **protocols.io** dx.doi.org/10.17504/protocols.io.mejc3cn
Published: 28 Dec 2017

Guidelines



Enlarge in New Window

Brief Overview:

Measurement of DNA can be done using two broad methods (Spectrophotometric or Fluorometric).

Qubit is a fluorometric method and uses a double stranded DNA specific dye. Therefore the sensitivity is higher and impurities like RNA or nucleotides in the sample will not affect the quantification as it would in spectrophotometric methods like a Nanodrop.

The Qubit is first calibrated using two standards. To make a measurement, 199ul of buffer +1ul of dye + 1ul of sample is added to a qubit tube, which is inserted into the instrument.

Here, we use OpenTrons to prepare samples for qubit measurement. It also saves time by requiring only one initial calibration for each container. It uses p1000 to distribute master mix and p10 to distribute unique samples.

Please see the flowchart of the protocol below.

Materials:

Qubit dsDNA HS Assay Kit (1)

Invitrogen catalog number: Q32851

Qubit Assay Tube (1 per sample)

Invitrogen catalog number: Q32856

Run Samples Placed in Strip Tubes

Instruments:

Qubit Fluorometer (1)

Invitrogen catalog number: Q32866

14ml Falcon Tube (1)

Alkali Scientific catalog number: CT5250

14ml Tube Rack (1)

96-well Qubit Tube Rack (1-2)

96-well Strip Tube Rack (1)

1000µl Tips

Fisher catalog number: 13-611-126

10µl Tips

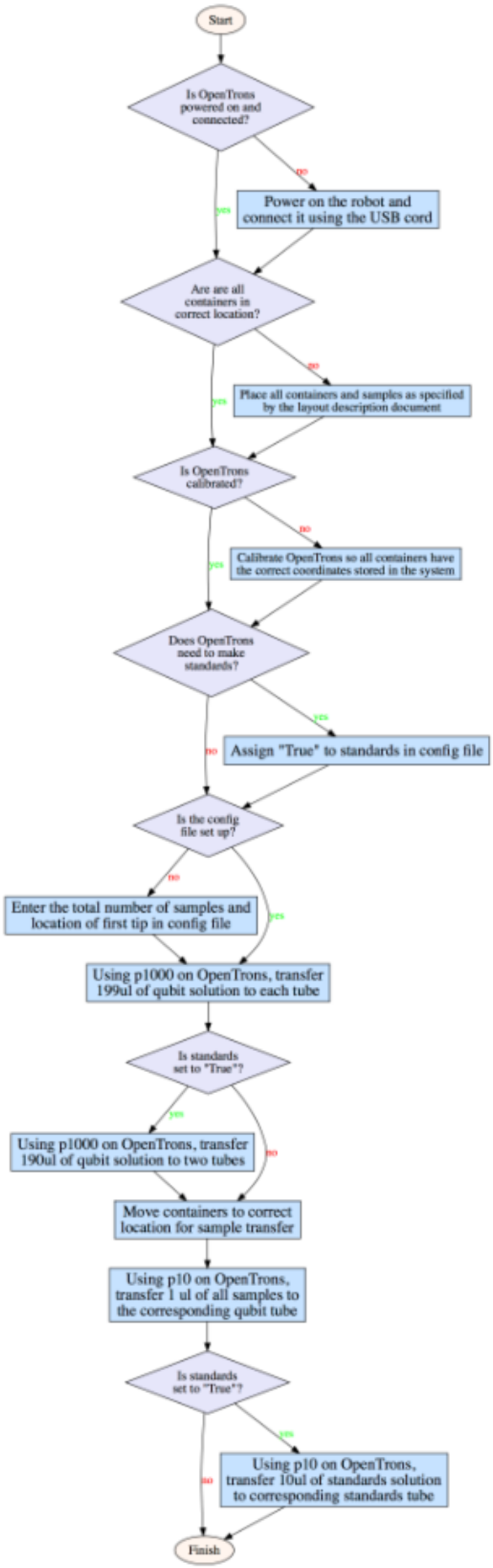
Eppendorf catalog number: 022491504

OpenTrons Pro (1)

Racks to hold tubes (Pictured in steps section)

Link to download graphviz code to generate this flowchart:

Graphviz Dot File



Materials

Qubit™ dsDNA HS Assay Kit Q32851 by Invitrogen - Thermo Fisher

Qubit™ Assay Tubes Q32856 by Invitrogen - Thermo Fisher

Qubit Fluorometer Q32866 by Invitrogen - Thermo Fisher

14ml Polystyrene Cell Culture Tubes CT5250 by Alkali Scientific

1000µL Basix™ Universal Pipette Tips 13-611-126 by Fisher Scientific

10µl Pipette Tips 022491504 by Eppendorf

OT-One Pro View by Opentrons

Protocol

Before Start

Step 1.

Before starting, please see the Guidelines for the flowchart and more information.

One Time Code Setup

Step 2.

Save the following file in a designated folder on your computer and name it config.ini

This is the configuration file you can edit to enter unique run information.

cmd COMMAND

```
[changeForRun]
sample_number: 24
first_tip: 0
make_standards: False

[defaultSettings]
start: 1
offset_d: (3, 3, -.4)

[qubit1]
aspirate_volume: 199
standard_volume: 190
tiprack: a1_p1000rack
master_mix: b1_15mltube
qubit_tubes: a2_qubit_tubes
overflow: b2_qubit_tubes
trash: a3_trash

[qubit2]
aspirate_volume: 1
standard_volume: 10
standard_location: A1
```

```
air_gap: 2
tiprack: e1_p10rack
samples: d1_strips
qubit_tubes: d2_qubit_tubes
overflow: e2_qubit_tubes
trash: d3_trash
standards: e3_p5tube
```

config.ini

One Time Code Setup

Step 3.

Save these two files in the same folder as config.ini

(must be saved as .py files)

These are the files that contain the code to run each the two separate steps of the protocol.

cmd COMMAND

```
# use p10 multichannel to distribute samples from strip tubes to qubit tubes filled with ma
ster mix
cpath = '/Users/opentronscomputer/Documents/opentrons/qubit/public/config.ini'
#string of path to folder with all files
imp_dir = '/Users/opentronscomputer/Documents/opentrons/qubit/public/'

import sys
sys.path.insert(1, imp_dir)
import configparser
config = configparser.ConfigParser()

from opentrons import containers, instruments, robot
from container_list import *
from helpers import *

def tube_container(number):
    """Returns the correct container function with the necessary defaults."""
    return correct_container(number, qubit_tubes, overflow, capacity=24)

#read the config file with the run info
config.read(cpath)

#total number of samples
sample_number = config.getint('changeForRun', 'sample_number')
#True if robot is needed to make standards
standard = config.getboolean('changeForRun', 'make_standards')
#volume to aspirate
aspirate_volume = config.getint('qubit2', 'aspirate_volume')
#row of first tip (first row is row 0, second row is row 1, etc.)
first_tip_row = config.getint('changeForRun', 'first_tip')
#volume of standard solution to aspirate
standard_volume = config.getint('qubit2', 'standard_volume')
#row where standards are located
stloc = config.get('qubit2', 'standard_location')
#due to bug that default offset is not (0, 0, 0) when run on the robot
offset_d = eval(config.get('defaultSettings', 'offset_d'))
#volume of air gap in ul, to minimize errors
gap = config.getint('qubit2', 'air_gap')
```

```
#if you need to start protocol in the middle, write sample number that you need to start on

#(example: if you need to start at sample number 4, change to 4)
start = config.getint('defaultSettings', 'start')

#make sure tip container has 4 tips per row, alternating a space in between with the first
column (A) empty

#define containers
tiprack = eval(config.get('qubit2', 'tiprack'))
samples = eval(config.get('qubit2', 'samples'))
qubit_tubes= eval(config.get('qubit2', 'qubit_tubes'))
overflow = eval(config.get('qubit2', 'overflow'))
trash = eval(config.get('qubit2', 'trash'))
standards = eval(config.get('qubit2', 'standards'))
#set robot deck to the deck that containers are initialized in, due to opentrons bug that i
nitializes separate decks
robot._deck = tiprack.get_parent().get_parent() #must be placed before pipette is defined
#define pipette
p10 = pipette10(trash=trash, tiprack=[tiprack])

for i in range(start, sample_number + 1, 4): #loop through as many times as there are sampl
es
    p10.pick_up_tip(tiprack[four_position(i)]) \
       .air_gap(gap) \
       .move_to((samples[skip_row_position(i)], qubit_sample_offset(i, offset_d))).aspirate
(aspirate_volume) \
       .dispense(aspirate_volume+gap, (tube_container(i)[skip_four_position(i)].bottom(5)))
 \
       .drop_tip() #pick up tip, aspirate sample, dispense in qubit tube, drop tip

if standard: #if robot is needed to make standards
    while i % 4 != 0:
        i+=1 #starts pipette on a new row
    p10.pick_up_tip(tiprack[four_position(sample_number+1)]) \
       .aspirate(standard_volume, standards[stloc]) \
       .dispense(standard_volume, (tube_container(i)[skip_four_position(i)].bottom(5))) \
       .drop_tip() #pick up tip, aspirate standard solution, dispense in qubit tube, drop t
ip
```

2sample.py p10 samples distribution

One Time Code Setup

Step 4.

At the top of the two python (.py) files, change the example cpath at the top of the page to the path of your config file written using quotes

(on Macs, right click on your config.ini file and click 'Get info'. Copy the text next to the Where description under the General section, and it will give you the path to the correct folder. Paste this over the example cpath in the code file and add config,ini to the end before the last quote)

(on PC, right click on your config.ini file and click 'Copy as Path'. Paste this over the example cpath)

It should look something like the example:

cpath = '/Users/opentronscomputer/Documents/opentrons/qubit/public/config.ini'

One Time Code Setup

Step 5.

In the same two files, change the example imp_dir right below cpath to the to the path of your folder with all 3 files. This should just be the above cpath without the config.ini at the end. It should look something like the example:

imp_dir = '/Users/opentronscomputer/Documents/opentrons/qubit/public/'

One Time Code Setup

Step 6.

Save the following file as helpers.py in the same folder as the previous 3 files.

This file contains functions that help the protocol run correctly.

cmd COMMAND

```
def correct_container(current_sample, original, overflow, capacity):
    """Given current sample number, will return the correct container if an overflow contai
ner exists."""
    if current_sample <= capacity:
        return original
    else:
        return overflow

def skip_row_four_well_position(current_sample, start_well=0):
    """
    Converts current sample number to corresponding well number on
    96-well plate, if there were 4 samples per row plus a skip row
    in between samples. Restarts when sample number is greater than 24.
    """
    if (current_sample + start_well) > 24:
        current_sample = current_sample - 24
    row = int(current_sample // 4.1)
    return (2 * current_sample - 1)  + (row * 8) + start_well

def four_position(current_sample, start=1):
    """
    Converts current sample number to string of the
    corresponding row if there were 4 samples per row.
    """
    row = current_sample // 4 + start
    return 'A' + str(row)

def skip_row_position(current_sample, start=1):
    """
    Converts current sample number to a string of the corresponding row
    if there were 8 samples per row, plus a skip row in between samples.
    """
    row = (current_sample // 8) * 2 + start
    return 'A' + str(row)

def qubit_sample_offset(current_sample, default):
    """
    Converts current sample number to a tuple that alternates
    between an offset of 1 sample for the first half of the row,
    or the default offset for the second half of the rows.
    """
```

```
    remainder = current_sample % 8
    if remainder <= 4:
        return (default[0] - 9, default[1], default[2])
    else:
        return default

def skip_four_position(current_sample, start=1):
    """
    Converts current sample number to a string of the corresponding row
    if there were 4 samples per row, plus a skip row in between samples.
    Restarts when there are greater than 12 rows.
    """
    row = 2 * (current_sample // 4) + start
    if row <= 12:
        return 'A' + str(row)
    else:
        return 'A' + str(row-12)
```

helpers.py

One Time Code Setup

Step 7.

Save the following file as container_list.py in the same folder as the previous 4 files.

This file lists all the containers and pipettes used in the protocol (containers pictured in robot setup section). If you want to use your own containers, you can add them to the bottom of this list following the format used in this file. Just make sure to change the container name to your new container in the config.ini file (make sure to test your new protocol beforehand as the helper functions are written with our containers in mind).

cmd COMMAND

```
from opentrons import containers, instruments

a1_p1000rack = containers.load(
    'tiprack-1000ul-chem', #container name in opentrons system
    'A1', #slot of container
    'a1p1000tiprack', #name to remember calibrations
)

a2_qubit_tubes = containers.load(
    '96-deep-well',
    'A2',
    'a2qubit_tubes',
)

a3_trash = containers.load(
    'point',
    'A3',
    'trash',
)

containers.create(
    '15ml-short-tube',
    grid=(1, 1),
    spacing=(20, 20),
    diameter=18,
```

```
    depth=105,
)

b1_15mltube = containers.load(
    '15ml-short-tube',
    'B1',
    'b1_15mltube',
)

b2_qubit_tubes = containers.load(
    '96-deep-well',
    'B2',
    'b2qubit_tubes',
)

d1_strips = containers.load(
    '96-PCR-tall',
    'D1',
    'sample_tubes',
)

d2_qubit_tubes = containers.load(
    '96-deep-well',
    'D2',
    'd2qubit_tubes',
)

d3_trash = containers.load(
    'point',
    'D3',
    'd3trash',
)

e1_p10rack = containers.load(
    'tiprack-10ul',
    'E1',
    'tiprack',
)

e2_qubit_tubes = containers.load(
    '96-deep-well',
    'E2',
    'e2qubit_tubes',
)

e3_p5tube = containers.load(
    '96-deep-well',
    'E3',
    'e3p5_tubes',
)

def pipette1000(trash=a3_trash, tiprack=[a1_p1000rack]):
    """Create pipette, tiprack must be in list form if included"""
    return instruments.Pipette(
    name="p1000",
    axis="b",
    min_volume=100,
    max_volume=1000,
    tip_racks=tiprack,
    trash_container=trash,
```

```
        channels=1,
    )

    def pipette10(tiprack=[e1_p10rack], trash=[d3_trash]):
        """Create pipette, tiprack must be in list form if included"""
        return instruments.Pipette(
            axis = 'a',
            max_volume = 10,
            min_volume = 0.5,
            tip_racks = tiprack,
            trash_container = trash,
            channels = 8,
            name = 'p10',
    )
```

container_list.py

Set Up OpenTrons Robot

Step 8.

Plug the OpenTrons into the usb port of the computer and turn on the robot (you should see a blue light).

Set Up OpenTrons Robot

Step 9.

Open the OpenTrons App in Computer.

Set Up OpenTrons Robot

Step 10.

Select the correct port in the drop down menu on the upper right hand corner of the app.

Set Up OpenTrons Robot

Step 11.

Click 'Yes' when asked to Home the robot.

Set Up OpenTrons Robot

Step 12.

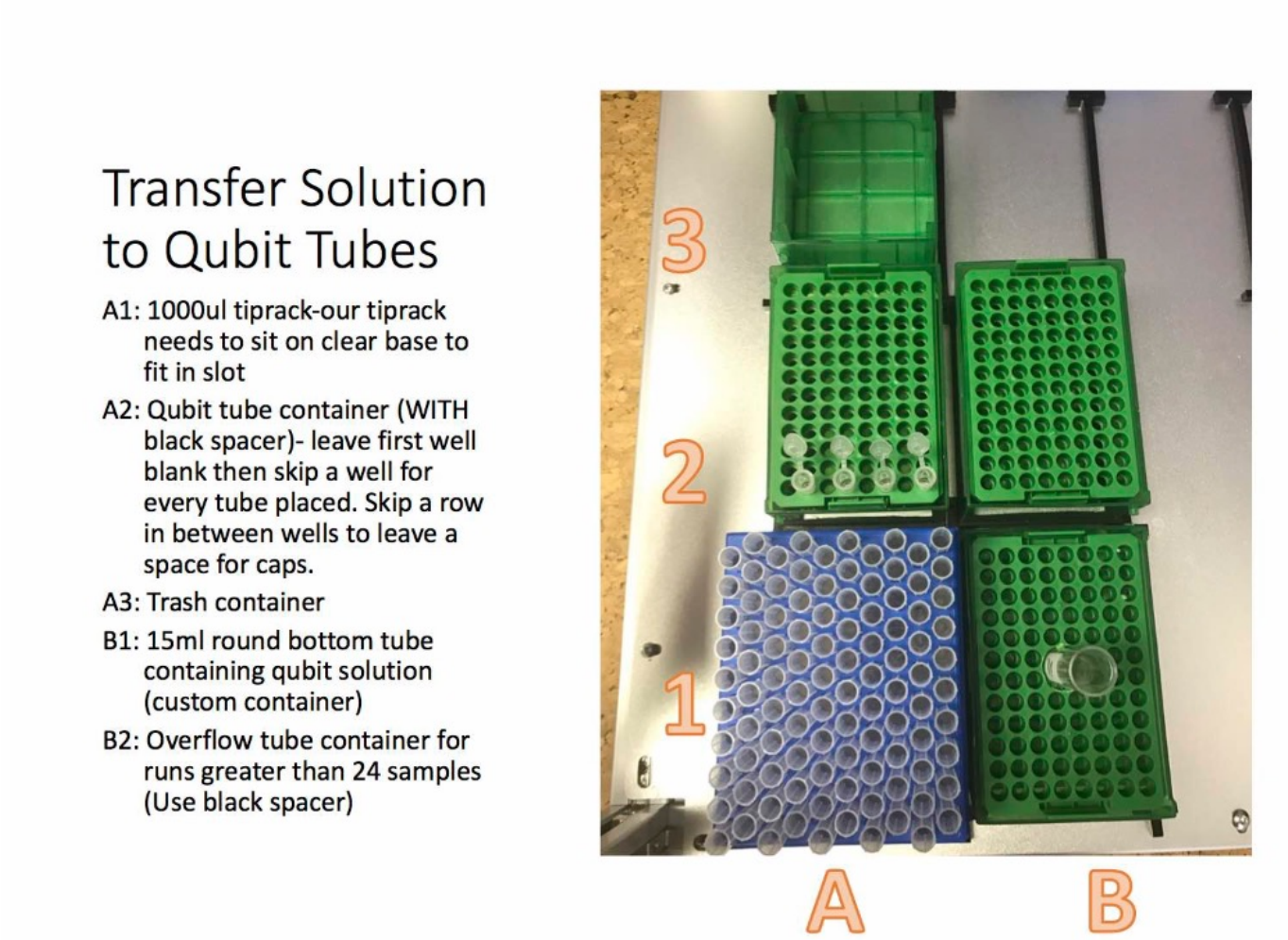
Set up containers according to this layout description.

(We use 96 well holders to hold qubit tubes as this should standardize spacing when using multichannel in second step of protocol. As for the 15ml tube rack, any rack that fits your large tube should work).

Important things to note:

- When arranging qubit tubes, leave the first column blank and start on the second column, skipping a column in between every tip/tube (see picture). Skip a row in between every tube to leave room for open caps.
- If OpenTrons is needed to make standards, set two empty qubit tubes in a new row behind the

last row of samples.



Enter Unique Run Details

Step 13.

Open the config.ini file and change the sample_number to the number of samples for your current run in the changeForRun section (not including standards).

Enter Unique Run Details

Step 14.

Change the first_tip to the number of tips that are already used in your tiprack (0 in a brand new tip box).

Enter Unique Run Details

Step 15.

If you would like OpenTrons to make standards, change make_standards equal to 'True', otherwise, set to 'False' (capitalize first letter).

Calibration

Step 16.

Save the edited config.ini file.

Upload the file containing the 1st step (1master_mix.py) onto the OpenTrons app.

Calibration

Step 17.

If this is the first time running the protocol, you will need to do a one-time container calibration. You can find OpenTrons instructions for how to calibrate containers and pipettes. Otherwise you only need to check calibrations if you feel it is necessary (i.e. first time running protocol in the day, if you feel that calibrations may have shifted). Here are some tips:

Click on the tiprack container and press 'Move to' for the pipette to move to the tiprack, see if pipette accurately fits the tip.

If tiprack calibration is accurate, the rest of the containers *usually* are correctly calibrated.

If calibration is off:

Z-axis is usually calibrated correctly, so move 20-40mm up, and slowly move back down the same amount changing X and Y to the correct location.

Press 'Save' when location is correct.

Check other containers:

Click on the other containers in the checklist while tip is on pipette, and click 'Move to'.

If calibration incorrect, calibrate just like how you calibrated the tiprack.

Repeat for all containers.

To drop tip:

Click on the tiprack container and press 'Move to'.

Raise the pipette 20mm up in the Z-axis and Click on the pipette in the checklist.

Find the Drop Tip section and click 'Move to'.

Qubit Solution Transfer

Step 18.

Click the 'Run' button to run the step.

When prompted to home, press OK.

This step uses the p1000 and transfers 199µl of qubit solution to each tube.

If standards is set to 'True', 190µl of qubit solution is transfered to the two tubes in the separate row in the end.

AMOUNT
199 µl Additional info: Qubit Solution
AMOUNT
190 µl Additional info: Qubit Solution

Set Up: Sample Transfer

Step 19.

If a different OpenTrons is needed for the next steps, home the current robot by pressing 'All' in the home section in the upper left corner of the OpenTrons App. Unplug the current robot and power off before repeating Steps 8-11 on the new robot to set up and power on the new robot.

GOTO
If different robot needed for 10µl multichannel -> go to step #8

Set Up: Sample Transfer

Step 20.

Set up containers according to this layout description

Important things to note

- When arranging tips , leave the first column blank and start on the second column, skipping a column in between every tip/tube (see picture)
- If OpenTrons is needed to make standards, place two extra tips in a new row behind the last row of samples, and set the container with the two standard solutions in slot E3 (see picture)

Transfer Samples to Qubit Tubes

D1: Sample Container- Samples in PCR strips- skip row between samples to leave room for caps

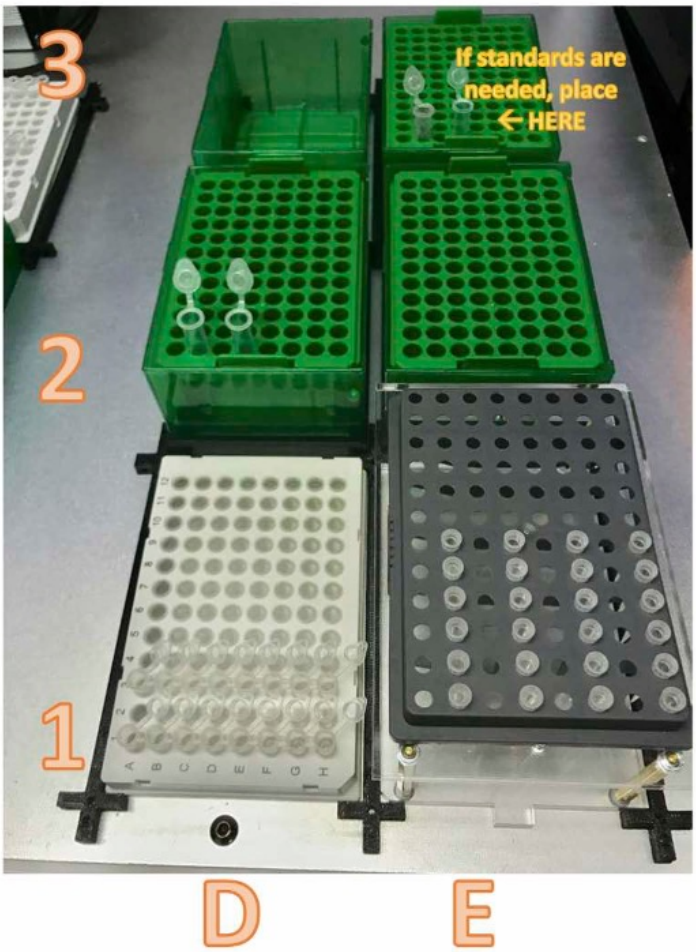
D2: Qubit tube container (WITH black spacer)- leave first well blank then skip a well for every tube placed. Skip a row in between wells to leave a space for caps. Has qubit solution from previous step.

D3: Trash Container

E1: 10ul tiprack- take out the tips in first, third, fifth, and seventh column

E2: Overflow tube container for runs greater than 24 samples

E3: If standards needed, place two 0.5ml tubes containing the standards in slot B1 and D1 of green container



Calibration: Sample Transfer

Step 21.

Edit and save the config file to the correct information for the second step

(should only need to change first_tip in the changeForRun section, this needs to be set to the number of rows to skip in your 10µl tiprack)

Upload the file containing the 2nd step (2samples.py) onto the OpenTrons app.

Repeat Step 17 for new containers to calibrate for p10.

GOTO

If p10 needs to be calibrated -> go to step #17

Transfer Samples to Qubit Tubes

Step 22.

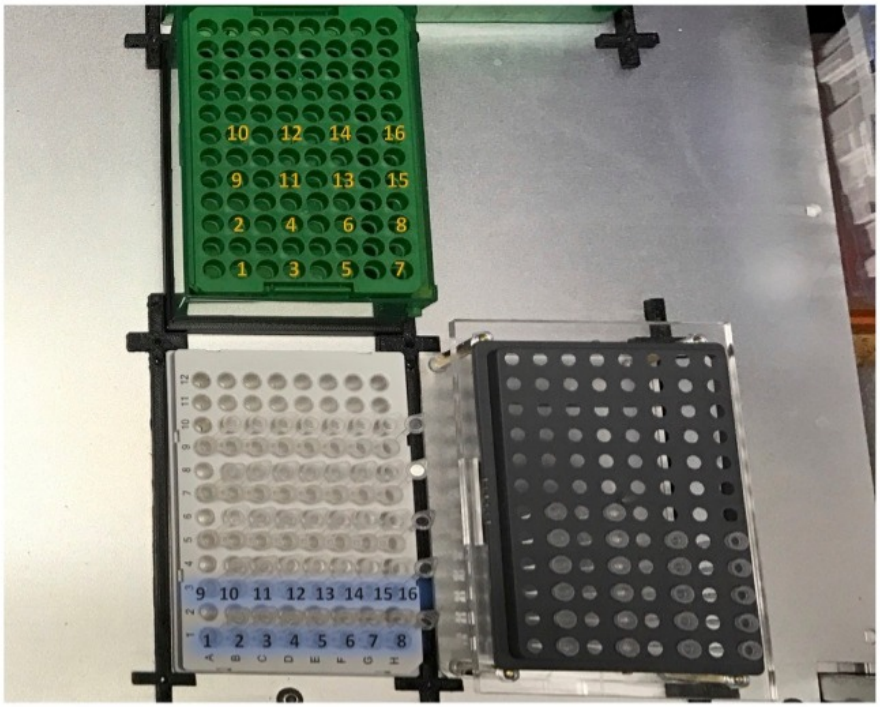
Click the 'Run' button to run the step.

When prompted to home, press OK.

This step uses the p10 and transfers 1µl of a sample to its corresponding tube.

(The robot transfers the odd number samples of the first row, to the first row of the qubit tubes, and then the even number samples to the next qubit tubes row. The odd number samples of the second row will then be transfered to the third row of the qubit tubes, followed by the even number samples to the forth row... see picture below)

If standards is set to 'True', the p10 transfers 10µl of the necessary standard solution to the standard tubes.



AMOUNT
1 µl Additional info: individual samples
AMOUNT
10 µl Additional info: standard soltuion

Finish

Step 23.

Home the current robot by pressing 'All' in the home section in the upper left corner of the OpenTrons App. Unplug the current robot and power off.