A Guide to Automating DNA Winner-Take-All Neural Network Validation Assays With Opentrons

Yaman Al Janaideh

Introduction

This guide provides detailed instructions for automating DNA Winner-Take-All (WTA) neural network validation assays using the Opentrons OT-2 robot [1]. It covers two key protocols and their corresponding assay templates:

- **WTA Memory Validation Protocol**: Validates individual memories (up to four per assay) within the DNA WTA neural network by running one of four assays—Multiplication, Summation, Annihilation, or Restoration. This protocol tests the individual subfunctions of the DNA WTA for each memory.
- **WTA Assembly Validation Protocol**: Validates the assembly and functioning of the complete DNA WTA neural network, testing the full cascade of five subfunctions (Multiplication, Summation, Annihilation, Restoration, Reporting) with all memories in the reaction. This protocol assesses the full network's output given each input.

Protocols can be found: here

Assay Templates: Each serves as the list of the components of the network the Opentrons will\be handling. They can all be found in Appendix A.

- **Memory Validation Templates** (4 templates):
  - Multiplication Assay Template
  - Summation Assay Template
  - Annihilation Assay Template
  - Restoration Assay Template
- **Assembly Validation Template** (1 template)

By following this guide, you can set up, customize, and run these protocols to accelerate your research and development efforts.

Note: The presented protocols were developed by our team. Given the open-source nature of the Opentrons API, please feel free to build upon the current versions and share them with the community.

[1] Opentrons, 2024. Opentrons lab automation lab robots for life scientists. https://opentrons.com/

Contents

Overview

Automating the validation of DNA WTA neural networks significantly improves throughput, accuracy, and reproducibility. The Opentrons OT-2 robot automates the preparation and mixing of DNA components required for these assays, handling complex liquid handling tasks that are impractical to perform manually at scale.

This guide will help you set up and run the WTA Memory Validation and WTA Assembly Validation protocols using the Opentrons OT-2 robot.

Prerequisites

Before you begin, ensure you have the following:

Equipment and Software

- **Opentrons OT-2 Robot** with the latest firmware.
- **Opentrons App** installed on your computer.
- **Python Protocol Scripts**
- **CSV Templates for Assay Setup** (provided in the Appendices):
- **Fluorescence Plate Reader** capable of reading 96-well (memory assays) or 48-well plates (assembly assays).
- **Ethernet Cable** to connect your computer to the OT-2 robot (if necessary).

DNA Components and Reagents

- **DNA Components** prepared at the appropriate concentrations (see Preparing the DNA Components).
- **Buffer:** 1x TAE/Mg^{2+} with 0.01% Tween.

Labware and Equipment Setup

The following labware and equipment are required for both protocols. Specific differences are noted where applicable.

Pipettes

- **Memory Validation Protocol:**
    - P300 Single-Channel Gen2 (Left Mount)
    - P300 Single-Channel Gen2 (Right Mount)
- **Assembly Validation Protocol:**
    - P1000 Single-Channel Gen2 (Left Mount)
    - P300 Single-Channel Gen2 (Right Mount)

Tip Racks

- **Memory Validation Protocol:**
    - Opentrons 96 Tip Rack 300 µL (Slots 7 and 8)
- **Assembly Validation Protocol:**

    - Opentrons 96 Tip Rack 1000 µL (Slot 7)
    - Opentrons 96 Tip Rack 300 µL (Slot 8)

Tube Racks

- **Opentrons 24-Tube Rack with Eppendorf 1.5 mL Safe-Lock Snap Cap Tubes:**
    - **Slot 1:** For memories 1 and 2
    - **Slot 4:** For memories 3 and 4 (if applicable)
- **Buffer Tube Racks:**
    - **Memory Validation Protocol:**
        - Opentrons 15-Tube Rack with Falcon 15 mL Conical Tubes (Slot 5)
    - **Assembly Validation Protocol:**
        - Opentrons 6-Tube Rack with Falcon 50 mL Conical Tubes (Slot 5)

Preparing the DNA Components

Ensure all DNA components are prepared at the appropriate working concentrations and volumes.

Concentrations

- The protocols assume adding 10 µL of each DNA component into the reaction to achieve working concentrations in 100 uL reactions (protocol 1) and 1000 uL reactions (protocol 2).
- **Working Concentrations (assuming 10 µL added to reaction):**

Component	**Concentration**
Input strand	1X (100 nM)
Weight gate	1X
Input fuel strand	2X
Intermediate product	1X
Summation gate	1X
Sum strand	1X
Annihilation gate	4X
Restoration gate	1X

Output strand	1X
Output fuel	2X
Reporter gate	2X

Volumes

- **Memory Validation Assays (100 µL reactions):**
    - Prepare at least 300 µL of each DNA component at 1 µM for 1X concentration.
    - Prepare around 15 mL of buffer.
- **Assembly Validation Assays (1000 µL reactions):**
    - Prepare at least 1000 µL of each DNA component at 10 µM for 1X concentration.
    - Prepare around 50 mL of buffer.

Labeling

- Clearly label all tubes with their contents and source positions as per the CSV templates. Their location on the rack matters and so you do not want to get them mixed up.

Buffer

- Use 1x TAE/Mg^{2+} with 0.01% Tween buffer.

General Protocol Setup

The following steps apply to both protocols:

Step 1: Edit the CSV Template

- Select the appropriate CSV template for your assay (see Appendices).
- Edit the CSV file to specify the DNA components, their volumes and their positions in the tube racks and save.

Step 2: Configure Protocol Parameters

- **Test Type:** Select the assay corresponding to your template.
- **Number of Memories:** Input the number of memories being tested.
    - Must correspond to the number of memories included in your CSV file.
- **Number of Replicates:** Choose between single or triplicate replicates.
- **Simulation Mode:** Set to *False* when running on the OT-2 robot.

Step 3: Set Up Labware on the OT-2

- **Load Labware:**
  - Place the required labware on the OT-2 deck as per the app's instructions.
  - For some assays you will need two 1.5 mL tube racks.
    - The first two memories in the CSV are assumed to be in Rack 1 (Slot 1).
    - Additional memories will be in Rack 2 (Slot 4).
  - **Annihilators** (Annihilation and Assembly Assays) are assumed to be on Rack 1.
- **Load Tips:**
  - Fill the tip racks starting from position A1 down the columns.
  - Tip racks should be filled from left to right before the protocol starts.
- **Labware Position Check:**
  - If performing a labware position check, ensure there is a tip or tube in position A1 of each rack before starting.

Step 4: Run the Protocol

- **Launch the Opentrons App:**
  - Ensure the OT-2 robot is turned on.
  - Turn on the HEPA filter if available.
  - Connect your computer to the robot via Ethernet cable.
- **Upload the Protocol:**
  - Click on the **Protocols** tab.
  - Upload the desired protocol script.
- **Upload the CSV File:**
  - Ensure the CSV file is in the same directory as the protocol script.
  - In the Opentrons App, click the three dots next to the protocol name and select "Show in Folder".
  - Add the CSV file to this folder.
  - Click the three dots again and select "Reanalyze".
- **Start the Run:**
  - Click on the protocol and select "Start Setup".
  - Click "Continue to Parameters".
  - Input the protocol parameters when prompted.
  - Place labware into the appropriate slots as per the labware placement map.
  - Perform a Labware Position Check if needed.
  - Click **"**Run" to start the protocol.
- **Completion:**
  - Once finished, carefully remove the assay plate.
  - Proceed promptly to fluorescence measurement as the reaction has already begun and the signal rises quickly.

Protocol 1: WTA Memory Validation

Understanding the Assays

The WTA Memory Validation Protocol allows you to validate the subfunctions of individual memories within the DNA WTA neural network by running one of four assays:

1. **Multiplication Assay (Test 1):**
    - **Purpose:** Validates the input and weight multiplication logic of individual memories.
    - **Conditions:**

Condition	Components Included
1	Reporter
2	Reporter, Output
3	Reporter, Restoration, Output Fuel, Summation, Weight, Input (100 nM)
4	Reporter, Restoration, Output Fuel, Summation, Weight, Input (20 nM)
5	Reporter, Restoration, Output Fuel, Summation, Weight, Input (100 nM), Input Fuel
6	Reporter, Restoration, Output Fuel, Summation, Weight, Input (20 nM), Input Fuel

Order of Plating: Reporter, Restoration, Summation, Weight, Output Fuel, Input Fuel, Input (100 nM), Input (20 nM), Output

2. **Summation Assay (Test 2):**
    - **Purpose**: Validates the summation of intermediate products of individual memories.
    - **Conditions:**

Condition	Components Included
1	Reporter

2	Reporter, Output
3	Reporter, Restoration, Output Fuel
4	Reporter, Restoration, Output Fuel, Summation, Product

Order of Plating: Reporter, Restoration, Summation, Output Fuel, Product, Output

3. **Annihilation Assay (Test 3):**
    - **Purpose:** Validates the annihilation of competing sums.
    - **Conditions:**

Condition	**Components Included**
1	Reporter X, Output X
2	Reporter X, Reporter Z, Restoration X, Restoration Z, Sum X
3	Reporter X, Reporter Z, Restoration Z, Annihilation, Sum X (100 nM), Sum Z (20 nM)
4	Reporter X, Reporter Z, Restoration Z, Annihilation, Sum X (100 nM), Sum Z (80 nM)
5	Reporter X, Reporter Z, Restoration Z, Annihilation, Sum X (20 nM), Sum Z (100 nM)
6	Reporter X, Reporter Z, Restoration Z, Annihilation, Sum X (80 nM), Sum Z (100 nM)

Order of Plating: Reporter X, Reporter Z, Restoration X, Restoration Z, Annihilation, Output X, Sum Z (80 nM), Sum Z (20 nM), Sum X (80 nM), Sum X (20 nM), Sum X (100 nM), Sum Z (100 nM).

4. **Restoration Assay (Test 4):**
    - **Purpose:** Validates the restoration of the winning sum for individual memories.
    - **Conditions:**

Condition	**Components Included**
1	Reporter

2	Reporter, Output
3	Reporter, Restoration, Sum (100 nM)
4	Reporter, Restoration, Sum (20 nM)
5	Reporter, Restoration, Output Fuel, Sum (100 nM)
6	Reporter, Restoration, Output Fuel, Sum (20 nM)

Order of Plating: Reporter, Restoration, Output Fuel, Output Sum (100 nM), Sum (20 nM)

Important Notes

- Default protocol assumes one input and one weight per memory.
- When testing more than two memories, you will need two 1.5 mL tube racks.
    - The first two memories in the CSV are in Rack 1 (Slot 1).
    - Additional memories are in Rack 2 (Slot 4).
- **Annihilation Assay Specifics:**
    - Assumes only two memories (X and Z).
    - You must choose two memories in the Opentrons App as the parameter for the number of memories.
    - All components should be listed as memory_number 1 in the csv so that all memories' components end up in the same wells (i.e. comparing conditions only not memories).

Protocol Setup and Execution

Follow the general protocol setup steps outlined in General Protocol Setup, ensuring you select the appropriate assay type and input the correct number of memories.

Proceed to run the protocol as described, and after completion, promptly transfer the assay plate for fluorescence measurement.

Protocol 2: WTA Assembly Validation

Understanding the Assay

The WTA Assembly Validation Protocol tests the full assembly and functionality of the DNA WTA neural network. It validates the cascade of all five subfunctions of all memories for each input.

- This protocol allows you to test up to four memories simultaneously.
- You can include fewer memories by editing the corresponding CSV file and selecting the appropriate parameters in the Opentrons App.

Purpose: Validate the complete DNA WTA neural network assembly.

Conditions:

Conditions	**Components Included**
1	Reporter
2	Reporter, Output
3	All Memories: Reporters, Restorations, Output Fuels, Annihilations, Summations, Weights, Input Fuels, Input

Order of Plating: Reporters, Restorations, Summations, Annihilations, Weights, Input Fuels, Output Fuels, Outputs, Input

Important Notes

- Default protocol assumes one input and one weight per memory.
- When testing more than two memories, you will need two 1.5 mL tube racks.
    - The first two memories in the CSV are in Rack 1 (Slot 1).
    - Additional memories are in Rack 2 (Slot 4).
- **Annihilators** are always placed on Rack 1. Must be memory 0 in the CSV file.
- **Buffer** is in the 50 mL Falcon tube rack (Slot 5).
- **Reaction Volume:** Assumes 1000 µL reactions.
- **Liquid Handling:** Although reactions are larger, the protocol still aspirates 10 µL from each DNA component, assuming appropriate concentrations.

Protocol Setup and Execution

Follow the general protocol setup steps outlined in General Protocol Setup, ensuring you select the assembly validation assay and input the correct number of memories.

Proceed to run the protocol as described, and after completion, promptly transfer the assay plate for fluorescence measurement.

Plating Logic

Plating Logic for Memory and Assembly Validation Assays

In both the Memory Validation Assay and Assembly Validation Assay, a key goal is to avoid premature triggering of reactions and ensure efficient, accurate liquid handling. By following a predefined plating order, both protocols minimize tip usage and optimize plating speed while ensuring proper distribution of all components.

Buffer Plating First

For both assays, the buffer is always plated first across all wells because the DNA components are added in small volumes (10 µL per component). This ensures a stable base in each well before any DNA components are added. Plating the buffer first helps maintain the correct reaction environment and helps ensure as much of the 10 uL goes into the reaction volume.

Common Plating Logic for Memory and Assembly Assays

- **Predefined Order to Avoid Premature Reactions**:
  The predefined order of liquid types ensures that critical components, such as inputs or outputs, are added last. This order prevents premature triggering of reactions, which makes it hard to catch the rising of the signals.
- **Minimizing Tip Changes for Efficiency**:
  Both assays are designed to minimize the number of tip changes, which makes the process more efficient and faster. Each liquid type is aspirated only once per memory, and tips are reused across all conditions that the liquid will be distributed into. This reduces the total number of tips needed without sacrificing accuracy.
- **General Plating Strategy**:
  - First, the system loops over each liquid type (or liquid type pair in the case of memory validation).
  - For a given liquid type, it loops over all memories being tested.
  - For each memory, the system loops over the conditions where the liquid needs to be added.
  - The liquid is plated across the required conditions for a memory using a single pipette tip, which is only replaced when moving to the next memory. This ensures efficient use of pipette tips while maintaining accuracy.

Memory Validation Assay Plating Logic

- **Plating by Liquid Type Pairs**:
  In the memory validation assay, two pipettes (P300) are used to handle liquid type pairs.

The first pipette plates the left component of the pair, and the second pipette plates the right component.

- **Plating Sequence**:
    - **By Liquid Type Pair**: The system handles each liquid type pair in turn. The left component of the pair is plated first, followed by the right component.
    - **By Memory**: After selecting the liquid pair, the system iterates over each memory, plating the liquid for one memory before moving to the next.
    - **By Condition**: Within a given memory, the liquids are plated into specific conditions (e.g., different concentrations or replicates), ensuring the correct setup for each assay condition.
- **Efficiency Through Reduced Tip Changes**:
  Since the maximum volume for each liquid type in a memory is typically 300 µL or less, the same pipette tip can be reused for all conditions in that memory. The tip is only changed when moving to the next memory, improving plating speed and efficiency.

Assembly Validation Assay Plating Logic

- **Single Liquid Types**:
  In the assembly validation assay, a single P300 pipette is used to handle individual liquid types. Unlike the memory validation assay, there is no pairing of liquid types, but the same strategy of type → memory → condition is followed.
- **Special Considerations**:
    - **Reporter**: The reporter is plated with a one-to-one mapping for conditions 1 and 2, but for condition 3, it follows a one-to-many mapping, ensuring all relevant wells receive the reporter strand.
        - One source to one destination
        - One source to many destinations
    - **Input**: The input strand follows a one-to-one mapping for condition 3.
    - **Output**: The output strand is plated with a one-to-one mapping for condition 2
- **Plating Sequence**:
    - **By Liquid Type**: Each liquid type is handled in turn, ensuring that all components are plated before the next one is added.
    - **By Memory**: For each liquid type, the system iterates over all memories being tested.
    - **By Condition**: Within each memory, the liquid type is plated into the appropriate conditions, just as in the memory validation assay.
- **Efficiency and Speed**:
  Similar to the memory validation assay, the maximum volume for each liquid type is 300 µL or less. This allows the same pipette tip to be reused across all conditions for a memory, minimizing tip changes and optimizing the overall plating process.

Data Analysis

After completing the assays:

- **Fluorescence Measurement:**
  - Immediately transfer the assay plate to a fluorescence plate reader.
  - Measure fluorescence intensity over time for each well according to your fluorophores (each memory corresponds to a fluorophore).
- **Data Interpretation:**
  - Compare fluorescence readings across conditions and replicates.
  - Analyze the data to validate the performance of individual memories or the assembled network.
  - Perform min-max normalization.
- **Documentation:**
  - Record all observations and data in your lab notebook or electronic data management system.
  - Save the run logs and associate them with your experimental data.

Troubleshooting and Best Practices

- **Tip Usage:**
  - Ensure tip racks are adequately stocked. The robot picks tips starting from position A1.
  - Tip racks should be filled from left to right before starting the protocol.
- **Labware Calibration:**
  - Perform a labware position check if you notice pipetting inaccuracies.
  - Place a tip or tube in position A1 of each rack before starting the check.
- **Liquid Handling:**
  - Verify that all liquids are prepared at the correct concentrations and volumes.
  - Ensure tubes are placed in the correct positions as specified in the CSV files.
  - The location of the buffer is referenced to the 15 mL tube rack in the Memory Validation Protocol and the 50 mL tube rack in the Assembly Validation Protocol.
  - DNA components (strands and gates) are referenced to the 1.5 mL tube racks in both protocols.
  - The protocol calculates the liquid height in tubes dynamically and updates it after each aspiration to ensure accurate pipetting.
- **Simulation Mode:**
  - Use simulation=True to test the protocol without running the robot.

        - We recommend using **Google Colab** for simulation (example in Appendix) .
        - Review the simulation log for any errors before actual execution.
    - Perform a dry run on the OT-2 without liquids to verify setup.
    - Conduct a test run with water before using actual reagents.
- **Assay Conditions:**
    - Conditions are plated along the columns, and replicates along the rows.
    - Memories are plated clockwise starting with the first memory in the CSV in the top-left corner (A1).
    - The protocol plates liquids by liquid type in a predefined order to avoid premature reactions.
        - Inputs, fuels, and output strands are plated last because they can trigger gates and start reactions or leak.
- **Fluorescence Readout:**
    - Be prompt in transferring the plate to the reader to prevent any degradation of the signal.
- **Data Management:**
    - Save the run log and upload it along with the corresponding data CSV file.
    - Update your experimental log accordingly.

Appendices

Appendix A: Template CSV Files

The CSV templates are structured to specify the DNA components, their types, source positions, and volumes.

Columns Description

- `memory_number`**:**
    - Numerical identifier for each memory (e.g., 1, 2, 3, 4).
    - Memory numbers are arbitrary IDs; use integers between 1 and 4.
    - For Annihilation Assays (Test 3) and Protocol 2:
        - **Annihilation** must be memory number `0`.
        - Use `b` for buffer entries.
- `liquid_type`**:**
    - Type of liquid component (e.g., reporter, restoration, input, annihilation, output_fuel, summation, weight).
    - For low concentrations, use suffixes like `_low` or `_lower` (e.g., `input_low`).
- `liquid_source`**:**
    - Source position in the tube rack (e.g., A1, B3).

    - **Buffers:**
        - Reference the 15 mL or 50 mL tube rack depending on the protocol.
    - **Liquids:**
        - Reference the 1.5 mL tube racks.
    - **Annihilators** are always on Rack 1.
- `liquid_total_volume`**:**
    - Total volume of the liquid in the tube (in µL).

Templates:

1. **Multiplication Assay (Protocol 1)**

memory_number	*liquid_type*	*liquid_source*	
b	buffer	B3	**12000**
1	restoration	A1	**1000**
1	reporter	A2	**1000**
1	summation	A3	**1000**
1	weight	A4	**1000**
1	output	A5	**1000**
1	output_fuel	A6	**1000**
1	input	B1	**1000**
1	input_low	B2	**1000**
1	input_fuel	B3	**1000**
2	restoration	C1	**1000**
2	reporter	C2	**1000**
2	summation	C3	**1000**
2	weight	C4	**1000**
2	output	C5	**1000**
2	output_fuel	C6	**1000**
2	input	D1	**1000**
2	input_low	D2	**1000**
2	input_fuel	D3	**1000**
3	restoration	A1	**1000**
3	reporter	A2	**1000**

3	summation	A3	**1000**
3	weight	A4	**1000**
3	output	A5	**1000**
3	output_fuel	A6	**1000**
3	input	B1	**1000**
3	input_low	B2	**1000**
3	input_fuel	B3	**1000**
4	restoration	C1	**1000**
4	reporter	C2	**1000**
4	summation	C3	**1000**
4	weight	C4	**1000**
4	output	C5	**1000**
4	output_fuel	C6	**1000**
4	input	D1	**1000**
4	input_low	D2	**1000**
4	input_fuel	D3	**1000**

2. Summation Assay (Protocol 1)

memory_number	*liquid_type*	*liquid_source*	*liquid_total_volume*
b	buffer	B3	**12000**
1	restoration	A1	**1000**
1	reporter	A2	**1000**
1	output	A3	**1000**
1	output_fuel	A4	**1000**
1	product	A5	**1000**
1	summation	A6	**1000**
2	restoration	B1	**1000**
2	reporter	B2	**1000**
2	output	B3	**1000**
2	output_fuel	B4	**1000**

2	product	B5	**1000**
2	summation	B6	**1000**
3	restoration	C1	**1000**
3	reporter	C2	**1000**
3	output	C3	**1000**
3	output_fuel	C4	**1000**
3	product	C5	**1000**
3	summation	C6	**1000**
4	restoration	D1	**1000**
4	reporter	D2	**1000**
4	output	D3	**1000**
4	output_fuel	D4	**1000**
4	product	D5	**1000**
4	summation	D6	**1000**

3. Annihilation Assay (Protocol 1)

memory_number	*liquid_type*	*liquid_source*	
b	buffer	B3	**12000**
1	restoration_x	A1	**1000**
1	reporter_x	A2	**1000**
1	sum_x	A3	**1000**
1	sum_lower_x	A4	**1000**
1	sum_low_x	A5	**1000**
1	output_x	A6	**1000**
1	annihilation	B1	**1000**
1	restoration_z	B2	**1000**
1	reporter_z	B3	**1000**
1	sum_z	B4	**1000**
1	sum_lower_z	B5	**1000**

1	sum_low_z	B6	**1000**

4. Restoration Assay (Protocol 1)

memory_number	*liquid_type*	*liquid_source*	*liquid_total_volume*
b	buffer	B3	**12000**
1	restoration	A1	**1000**
1	reporter	A2	**1000**
1	sum	A3	**1000**
1	sum_low	A4	**1000**
1	output	A5	**1000**
1	output_fuel	A6	**1000**
2	restoration	B1	**1000**
2	reporter	B2	**1000**
2	sum	B3	**1000**
2	sum_low	B4	**1000**
2	output	B5	**1000**
2	output_fuel	B6	**1000**
3	restoration	C1	**1000**
3	reporter	C2	**1000**
3	sum	C3	**1000**
3	sum_low	C4	**1000**
3	output	C5	**1000**
3	output_fuel	C6	**1000**
4	restoration	D1	**1000**
4	reporter	D2	**1000**
4	sum	D3	**1000**
4	sum_low	D4	**1000**
4	output	D5	**1000**
4	output_fuel	D6	**1000**

5. Assembly Assay (Protocol 2)

memory_number	*liquid_type*	*liquid_source*	
b	buffer	B3	**50000**
0	annihilation	B3	**1000**
0	annihilation	B4	**1000**
0	annihilation	B5	**1000**
0	annihilation	B6	**1000**
0	annihilation	D3	**1000**
0	annihilation	D4	**1000**
1	restoration	A1	**1000**
1	reporter	A2	**1000**
1	summation	A3	**1000**
1	weight	A4	**1000**
1	output	A5	**1000**
1	output_fuel	A6	**1000**
1	input	B1	**1000**
1	input_fuel	B2	**1000**
2	restoration	C1	**1000**
2	reporter	C2	**1000**
2	summation	C3	**1000**
2	weight	C4	**1000**
2	output	C5	**1000**
2	output_fuel	C6	**1000**
2	input	D1	**1000**
2	input_fuel	D2	**1000**
3	restoration	A1	**1000**
3	reporter	A2	**1000**

3	summation	A3	**1000**
3	weight	A4	**1000**
3	output	A5	**1000**
3	output_fuel	A6	**1000**
3	input	B1	**1000**
3	input_fuel	B2	**1000**
4	restoration	C1	**1000**
4	reporter	C2	**1000**
4	summation	C3	**1000**
4	weight	C4	**1000**
4	output	C5	**1000**
4	output_fuel	C6	**1000**
4	input	D1	**1000**
4	input_fuel	D2	**1000**

Appendix B: Simulations

How to run it in Colab / Jupyter notebook:

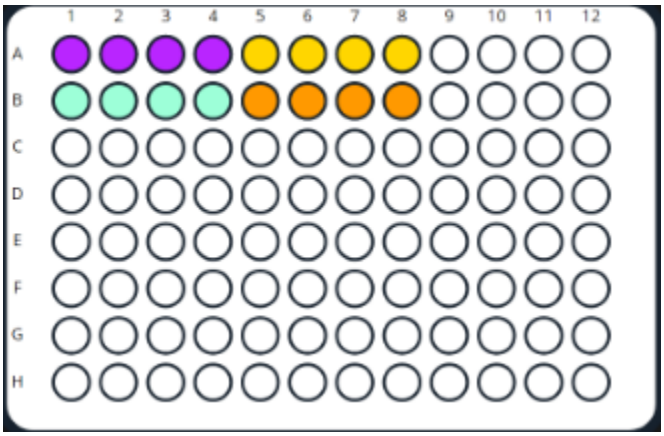
1. Run:

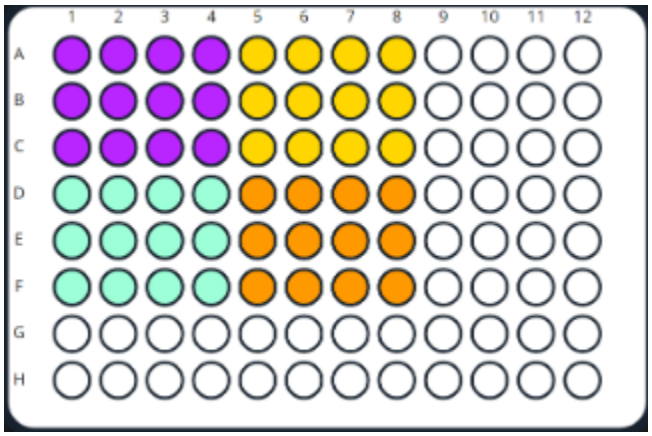
```
!pip install opentrons
```

2. Upload protocol and CSV into the same directory
3. Edit protocol params to set simulation to True and choose appropriate params.
4. Run:

```
!opentrons_simulate WTA_assembly_validation.py
```

Appendix C: Example of Summation Assay Single vs Triple replicate plating





Memory	Fluorophore
1	cy3.5
2	cy5.5
3	cy3
4	**cy5**

Note: We encourage you to build upon these protocols and share your improvements with the community.