

Bpod Tutorials

Updated 8/12/24 (Ksussman)

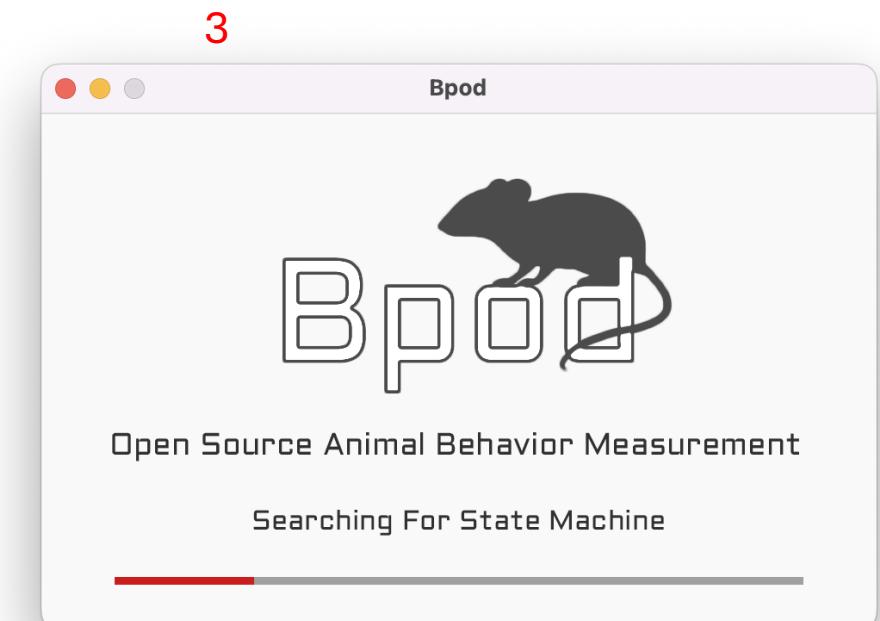
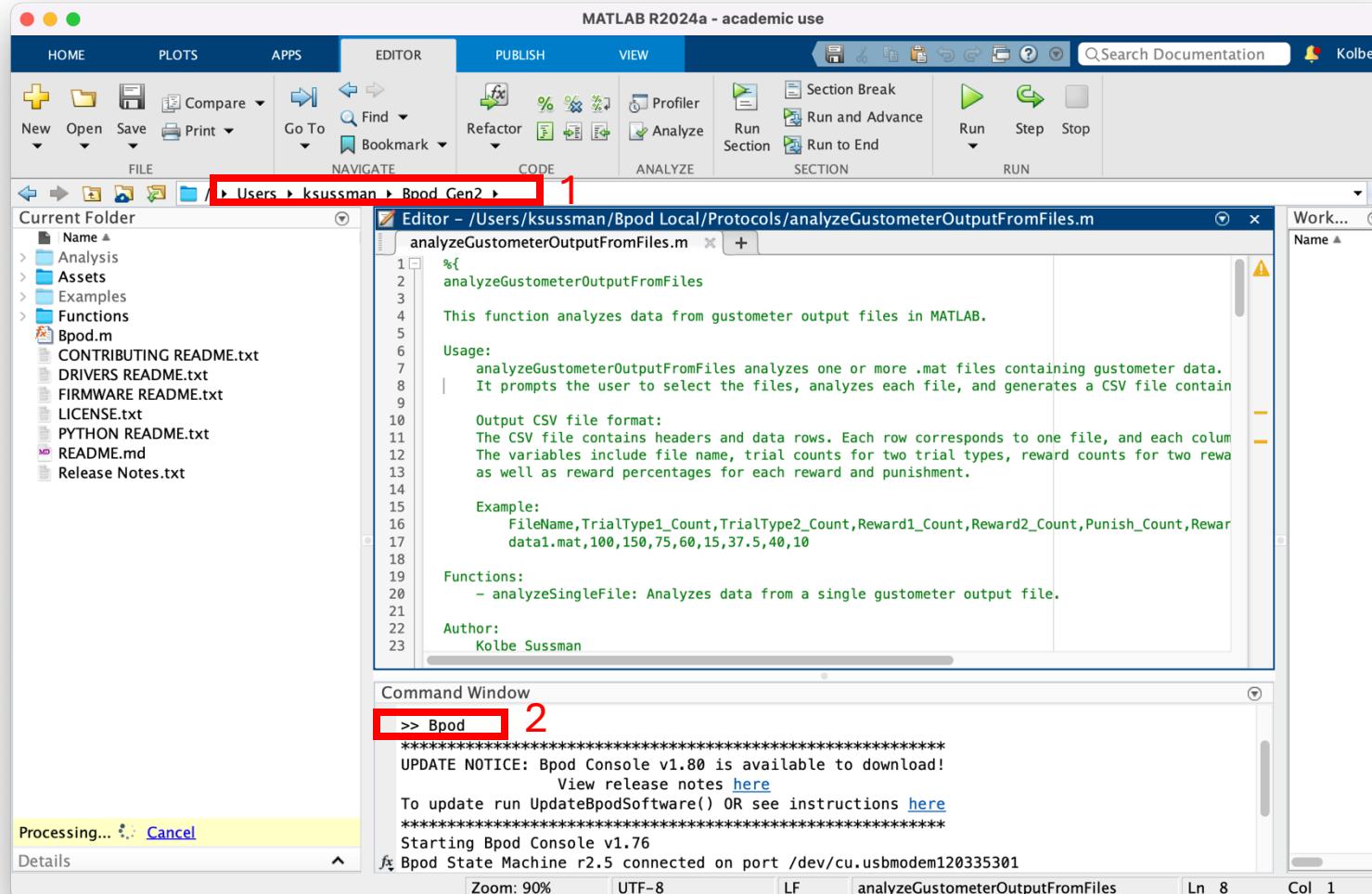
Contents:

1. [Using Bpod](#)
 - [Advance/Rgress Criteria](#)
2. [Analyzing Bpod Data Files](#)
3. [Understanding Bpod Protocol Scripts](#)

Using Bpod

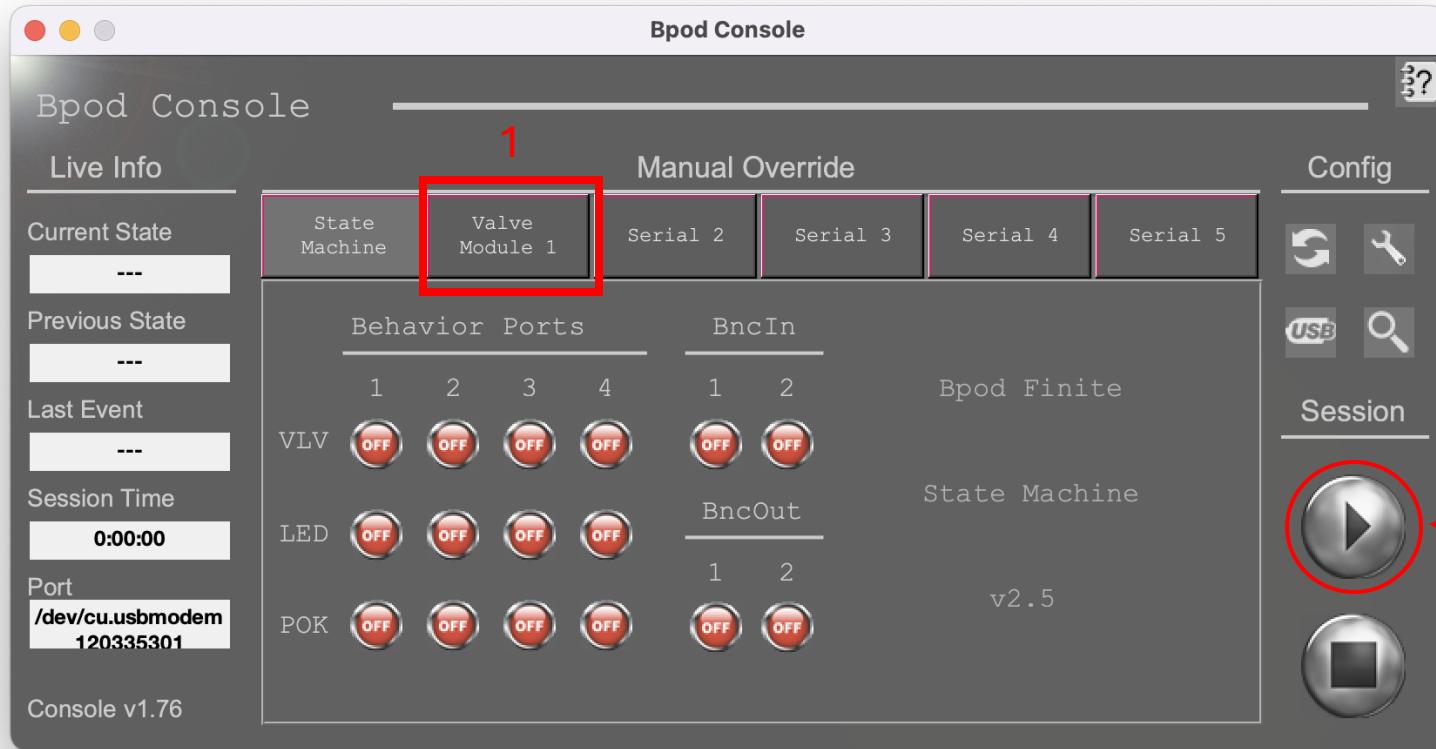
Required:

- Matlab ([Download here](#), EMU Students get a free download on one personal device)
- Bpod Software ([Download Here](#); the first time you download it it will ask you to setup, just select all default options)
- Gustometer Protocols ([Download Here](#))



Launch MATLAB

Set the working Directory to 'Users/Public/Documents/Bpod_Gen2-Master' (1) and Type Bpod into the Command Window (2) to launch Bpod Program (3)



To run a protocol, press the play button.

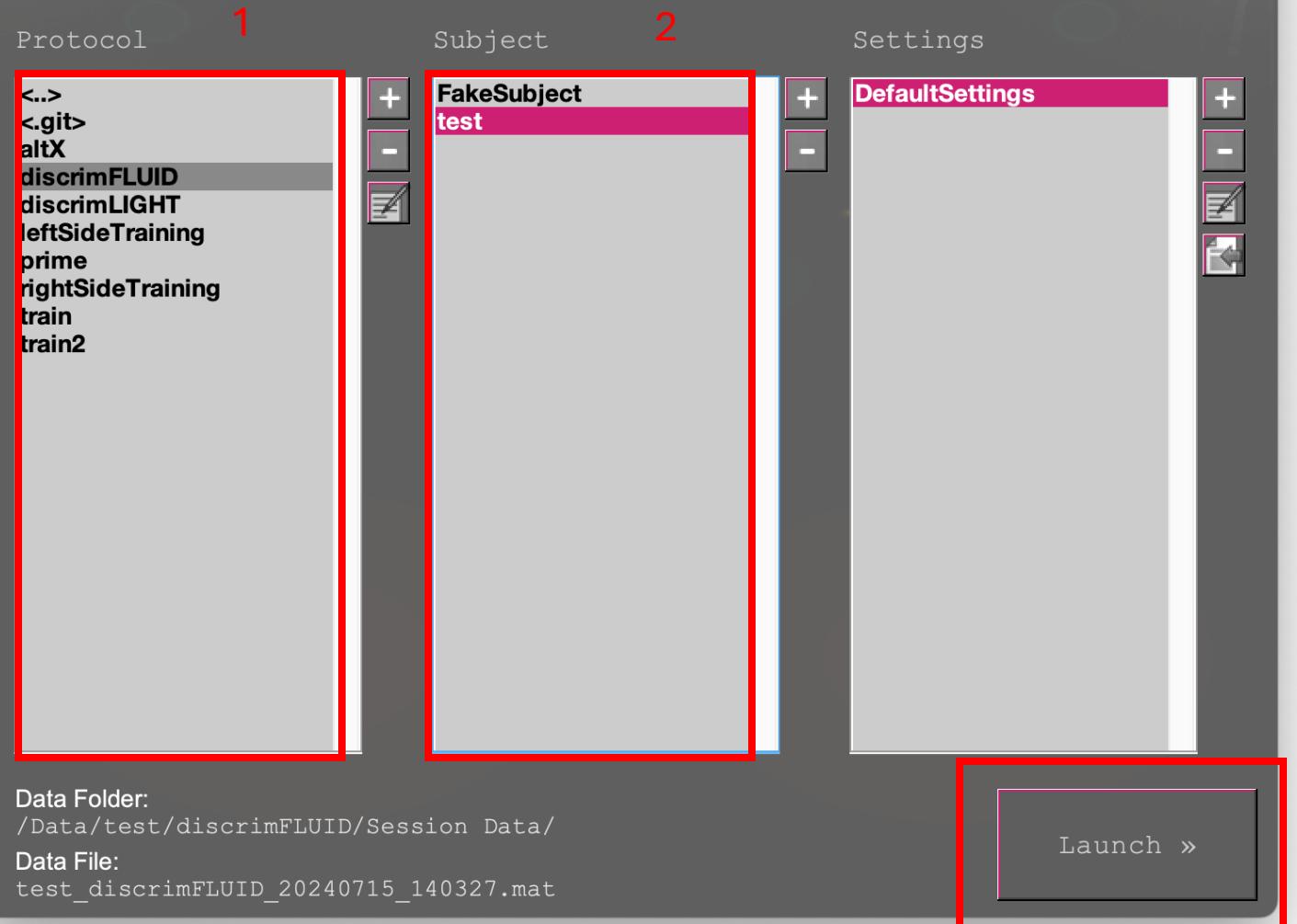
Once the Bpod program is running it should look like this.

IMPORTANT: If it does not list the 'Valve Module 1' (1), something is wrong. Check that everything is plugged in correctly and restart the Bpod program. If that doesn't work, there is probably something wrong with the circuit board. The circuit board can be cleaned using DI water, followed by 100% alcohol, then dried using the air compressor and left to completely dry for at least one hour.



Launch Manager

Protocol Launch Manager



This window should pop up when you click play.

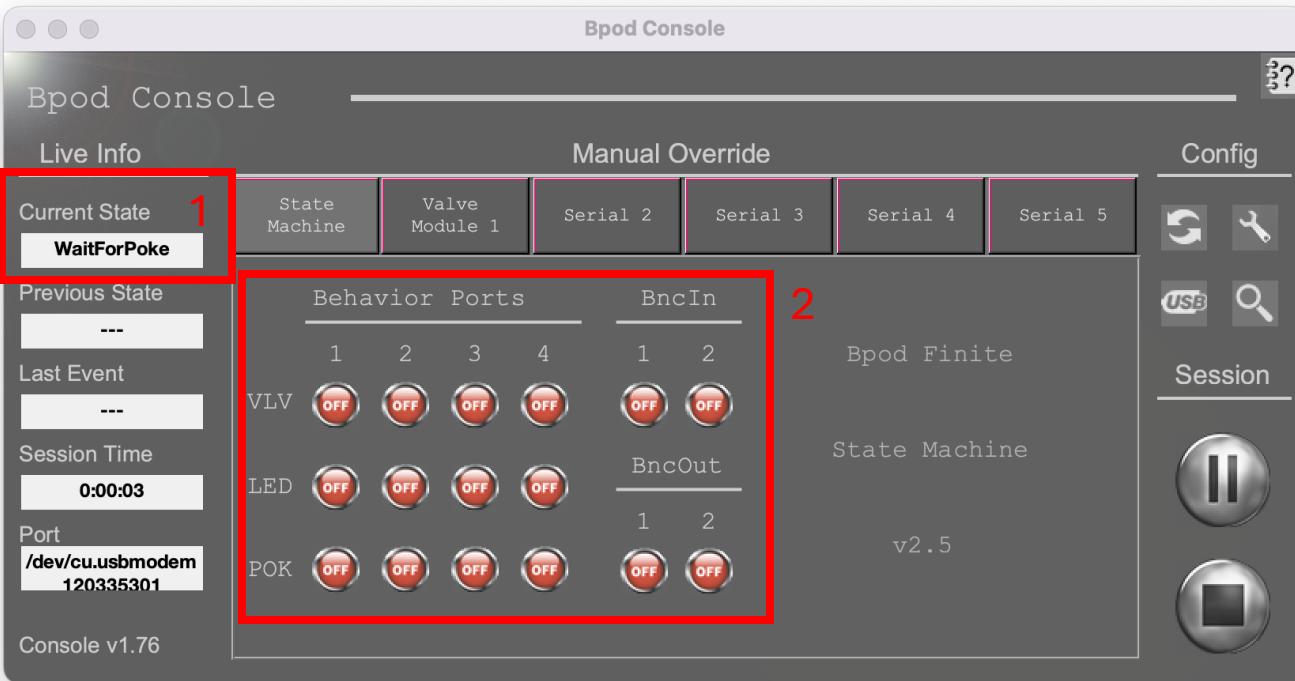
Select the Protocol you want to run (1)

Select the animal you are about to run (2)*

Keep 'DefaultSettings' selected

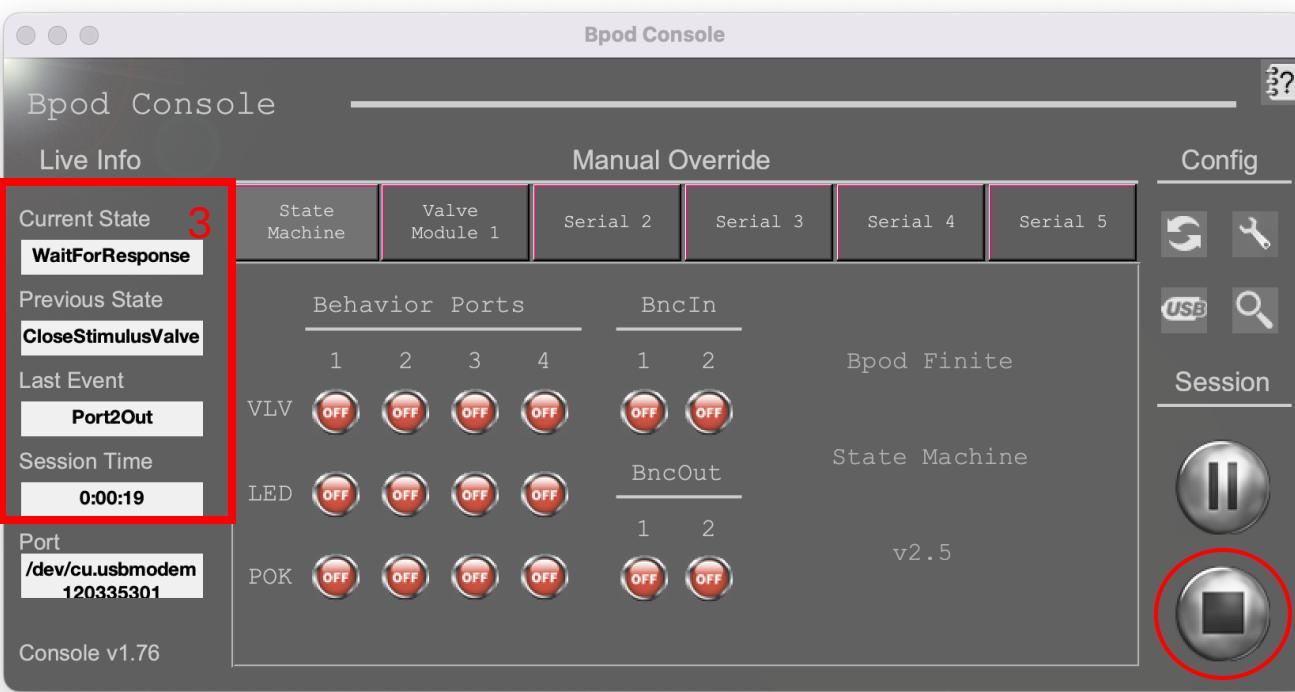
Click 'Launch >>' (3)

***IMPORTANT:** The first time you run an animal through a given protocol, IT WILL NOT SAVE THE DATA. If you are running an animal in a protocol for the first time, launch the program then immediately stop it. The next time(s) you run an animal in that protocol it will be fine. *This applies to EVERY new Protocol/Subject Combo.* This is a [known issue](#). It can probably be fixed but I don't have time



When you press ‘Launch’ the previous window will close automatically. The ‘Current State’ should always be ‘WaitForPoke’ (1) If it is not SOMETHING IS WRONG.

When you are running a mouse DO NOT click on any on/off buttons (2)



When animals are in the middle of a trial, the ‘Current State’, ‘Previous State’, ‘Last Event’, and ‘Session Time’ will all change (3). This is normal.

When you are done running an animal, hit the stop button to end the session.

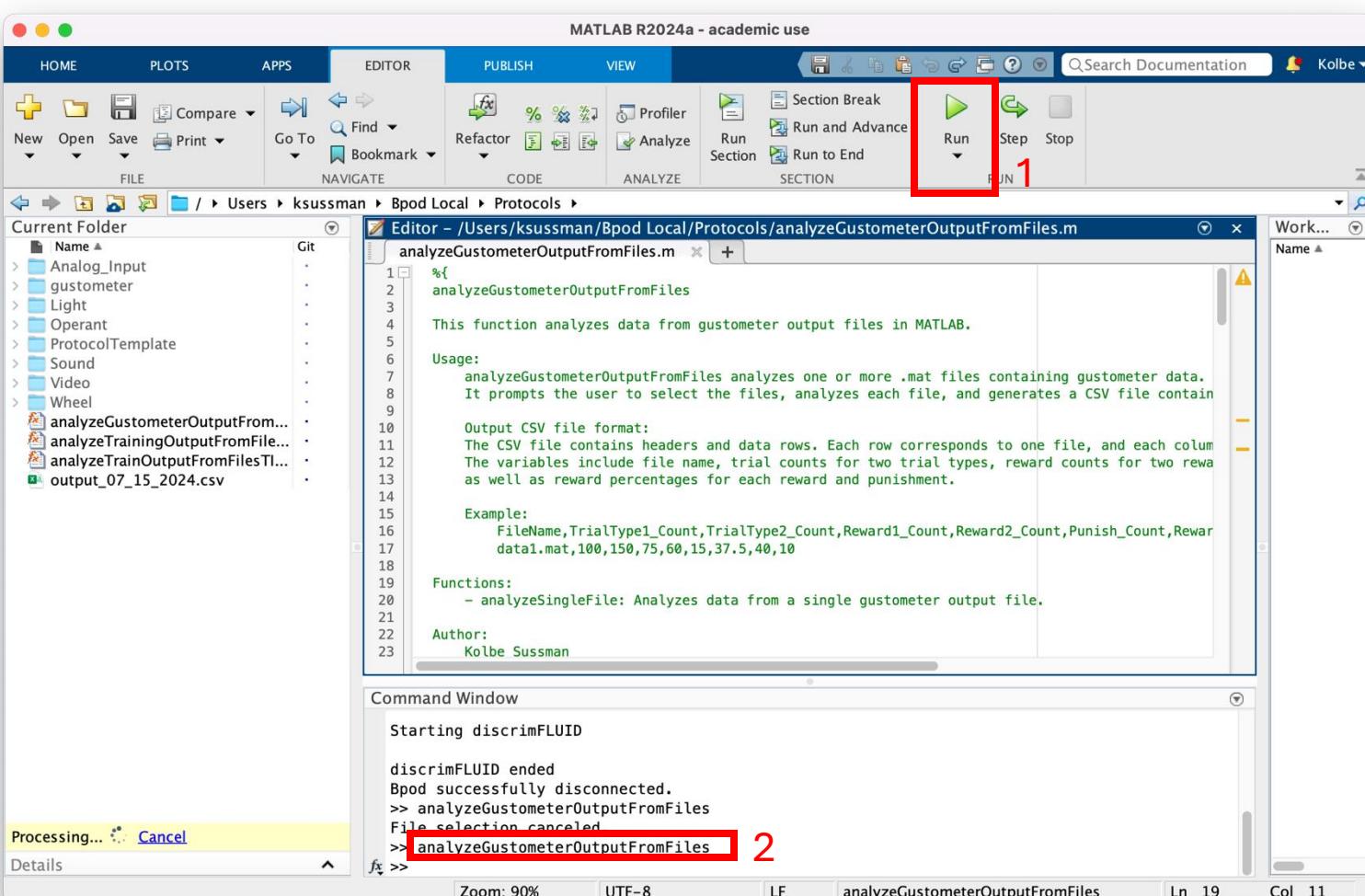
Important Information

- Order of protocols: train, train2, rightSideTraining, leftSideTraining, AltX (9), AltX (6), AltX (3), discrimFLUID, discrimLIGHT
- Advance/Rgress Criteria:
 - train, train2: Advance to next protocol after animal consistently gets 50 or more trials
 - rightSideTraining, leftSideTraining, AltX (9), AltX (6), AltX (3), discrimFLUID : Advance to next protocol after animal performs 80% or better. Rgress if animal gets 60% or worse for 3 sessions in a row.
 - Note: The ‘AltX’ protocol will ask you to input the number of trials before a switch when you run it. Please input the number in the parenthesis for each ‘AltX(x)’ step listed above.
- Length: Limit to no more than 2 straight weeks on H2O-deprivation. If you have to do more tests, give them weekends off. If you give an animal the weekend off, repeat most recent protocol, regardless of if advance criteria was met, on Monday as a “booster training”. This process will most likely take more than 2 weeks but can vary based on animal’s performance

Analyzing Bpod Data Files

Required:

- Matlab ([Download here](#), EMU Students get a free download on one personal device)
- Analysis Protocols ([Download Here](#))



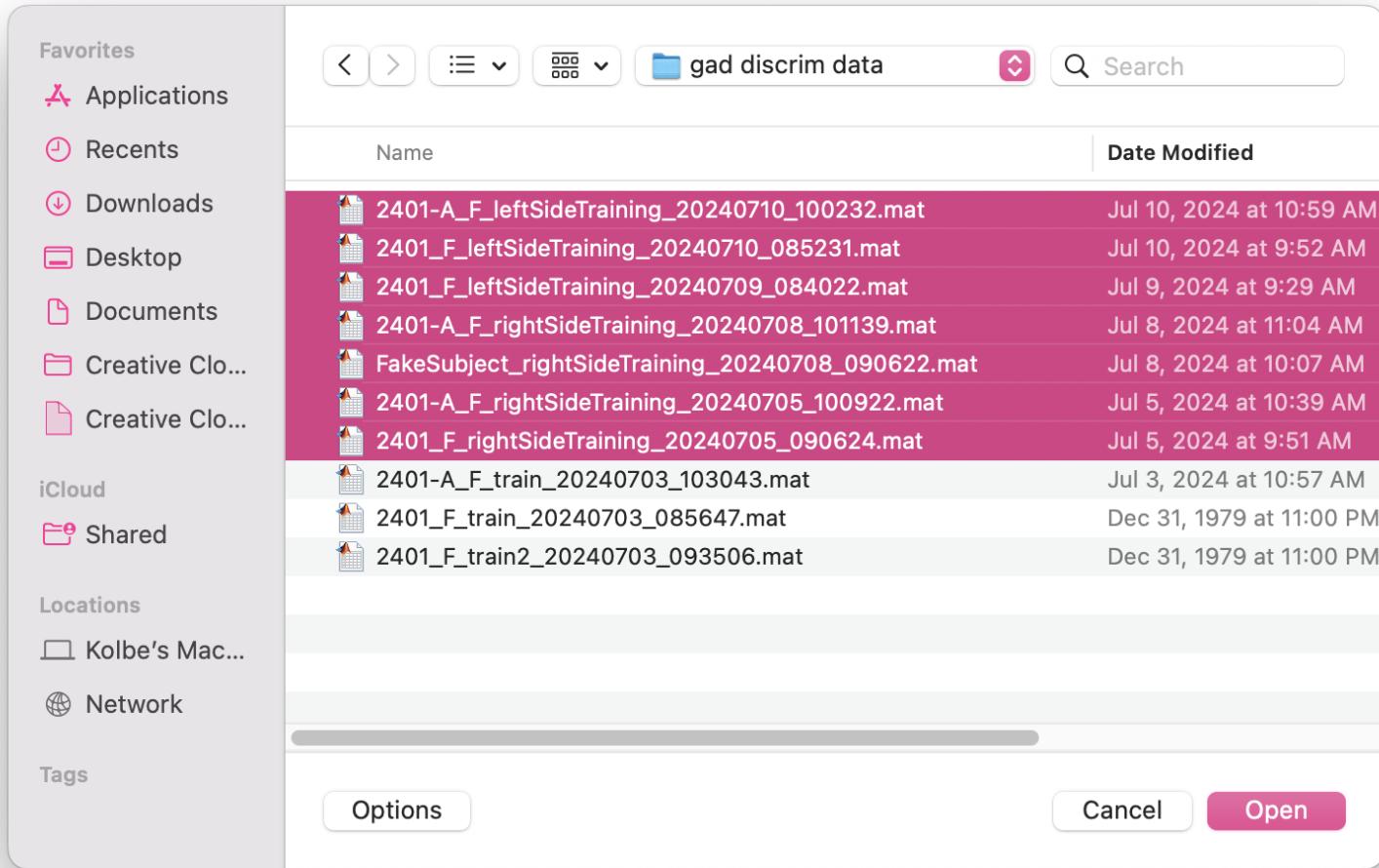
Launch MATLAB

Open the matlab script you want to use:

- `analyzeGustometerOutputFromFiles.m` is for Left/Right Side training, AltX, and discrimination protocols data
- `analyzeTrainingOutputFromFiles.m` is for train and train2 protocol data

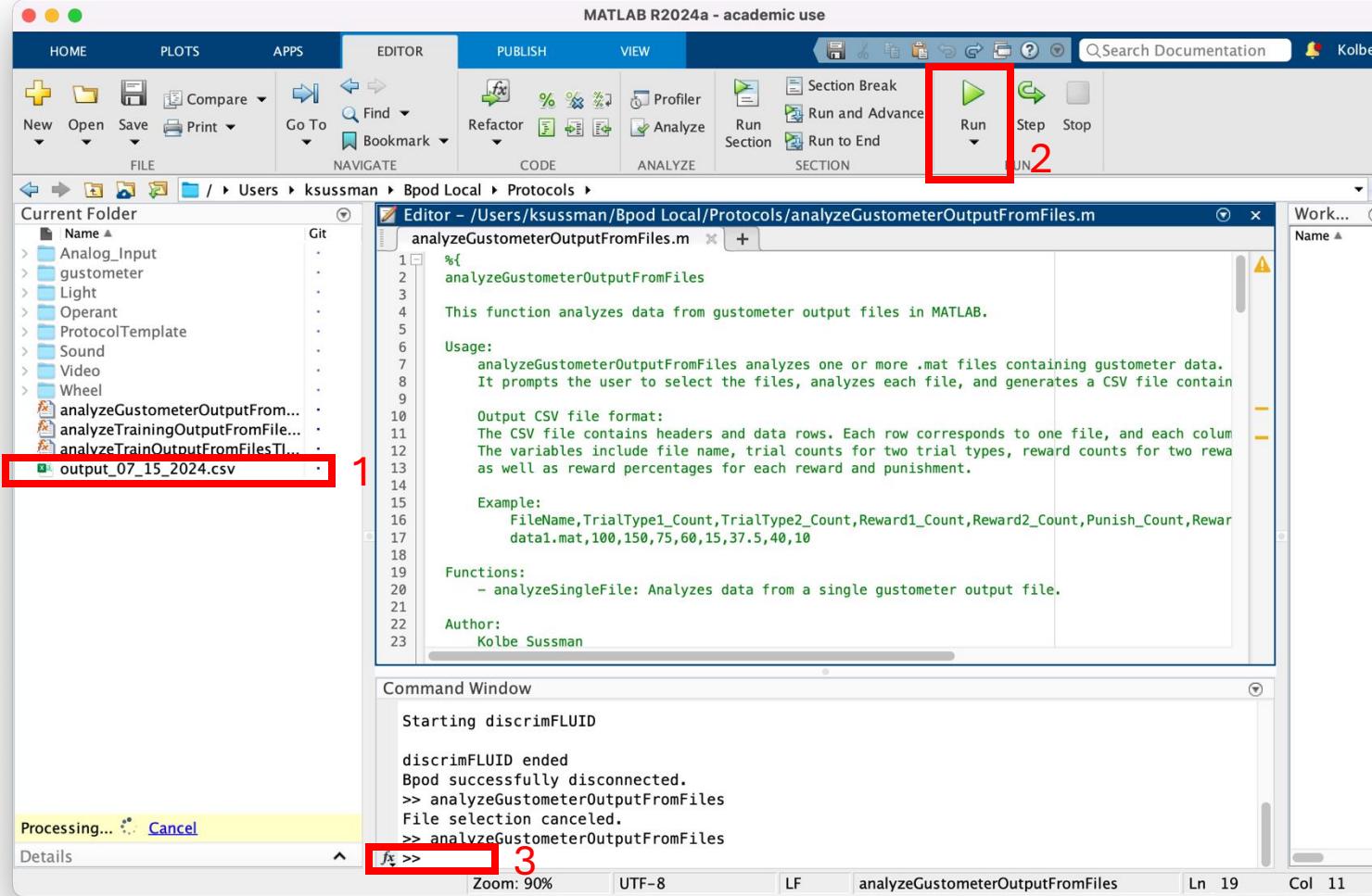
Note: I store these in the '/Bpod Local/Protocols' folder

Press 'Run' (1) or type the function name into the Command Window (2)



When you run the program, it should prompt you to select the data you want to analyze. If you want to select more than one file at a time, you will need to ensure all files are in the same folder before running the protocol.

Select the file(s) you want to analyze and click 'Open'



There are several signs the program has completed running. The output file will appear in the current folder (1), In the ‘Pause’ Button will be replaced with the ‘Run button again (2), and the command window will allow you to run code again, as indicated by ‘>>’ (3)

Import - /Users/ksussman/Bpod Local/Protocols/output_07_15_2024.csv

IMPORT **VIEW**

Column delimiters: Comma Range: A2:I8 Output Type: Table UNIMPORTABLE CELLS Import Selection Import

Delimited Fixed Width Delimiter Options Variable Names Row: 1 Text Options IMPORT

DELIMITERS SELECTION IMPORTED DATA

output_07_15_2024.csv

| | A | B | C | D | E | F | G | H | I | | |
|-----|---------------------------------|--------------|--------------|-------------------|------------|-------------|-------------|-------------|-------------------|---|---|
| | output07152024 | | | | | | | | | | |
| | FileName | TrialType... | TrialType... | Reward1... | Reward2... | Punish_C... | Reward1... | Reward2... | Punish_P... | | |
| 1 | Text | Number | Number | Number | Number | Number | Number | Number | Number | | |
| 2 | 2401-A... | 40 | 0 | 36 | 0 | 4 | 90 | 0 | 10 | | |
| 3 | 2401_F_I... | 43 | 0 | 37 | 0 | 6 | 86.0465... | 0 | 13.9534... | | |
| 4 | 2401_F_I... | 35 | 0 | 25 | 0 | 10 | 71.4285... | 0 | 28.5714... | | |
| 5 | 2401-A... | 32 | 0 | 26 | 0 | 6 | 81.25 | 0 | 18.75 | | |
| 6 | FakeSubj... | 58 | 0 | 51 | 0 | 7 | 87.9310... | 0 | 12.0689... | | |
| 7 | 2401-A... | 25 | 0 | 15 | 0 | 10 | 60 | 0 | 40 | | |
| 8 | 2401_F_r... | 36 | 0 | 22 | 0 | 14 | 61.1111... | 0 | 38.8888... | | |
| Q20 | X | V | fx | output_07_15_2024 | | | | | | | |
| | A | B | C | D | E | F | G | H | I | J | K |
| 1 | FileName | TrialType1_C | TrialType2_C | Reward1_Co | Reward2_Co | Punish_Coun | Reward1_Per | Reward2_Per | Punish_Percentage | | |
| 2 | 2401-A_F_leftSideTraining_20240 | 40 | 0 | 36 | 0 | 4 | 90 | 0 | 10 | | |
| 3 | 2401_F_leftSideTraining_202407 | 43 | 0 | 37 | 0 | 6 | 86.0465116 | 0 | 13.9534884 | | |
| 4 | 2401_F_leftSideTraining_202407 | 35 | 0 | 25 | 0 | 10 | 71.4285714 | 0 | 28.5714286 | | |
| 5 | 2401-A_F_rightSideTraining_2024 | 32 | 0 | 26 | 0 | 6 | 81.25 | 0 | 18.75 | | |
| 6 | FakeSubject_rightSideTraining_2 | 58 | 0 | 51 | 0 | 7 | 87.9310345 | 0 | 12.0689655 | | |
| 7 | 2401-A_F_rightSideTraining_2024 | 25 | 0 | 15 | 0 | 10 | 60 | 0 | 40 | | |
| 8 | 2401_F_rightSideTraining_20240 | 36 | 0 | 22 | 0 | 14 | 61.1111111 | 0 | 38.8888889 | | |

The columns are as follows:

- File Name (self-explanatory) – you can use regex in your analysis software of choice to pull out the animal, protocol, date, etc. or you can modify the MATLAB code to do it for you if you're feeling ambitious.
- TrialType1_Count – Count of trial type 1 (right side reward)
- TrialType2_Count – Count of trial typ1 2 (left side reward)
- Reward1_Count – count of how many times the right side was rewarded
- Reward2_Count – count of how many times the left side was rewarded
- Punish1_Count – count of how many times the animal was punished after delivering “right side” stimulus
- Punish2_Count – count of how many times the animal was punished after delivering “left side” stimulus
- TimeOut_Count – count of how many times the animal waited more than 15 seconds to respond to stimulus
- Reward1_Percentage – Percentage of how many trials the right side was rewarded
- Reward2_Percentage Percentage of how many trials the left side was rewarded
- Punish1_Percentage – Percentage of how many trials the animal was punished after delivering “right side” stimulus
- Punish2_Percentage – Percentage of how many trials the animal was punished after delivering “left side” stimulus
- TimeOut_Percentage – Percentage of trials the animal waited more than 15 seconds to respond to stimulus

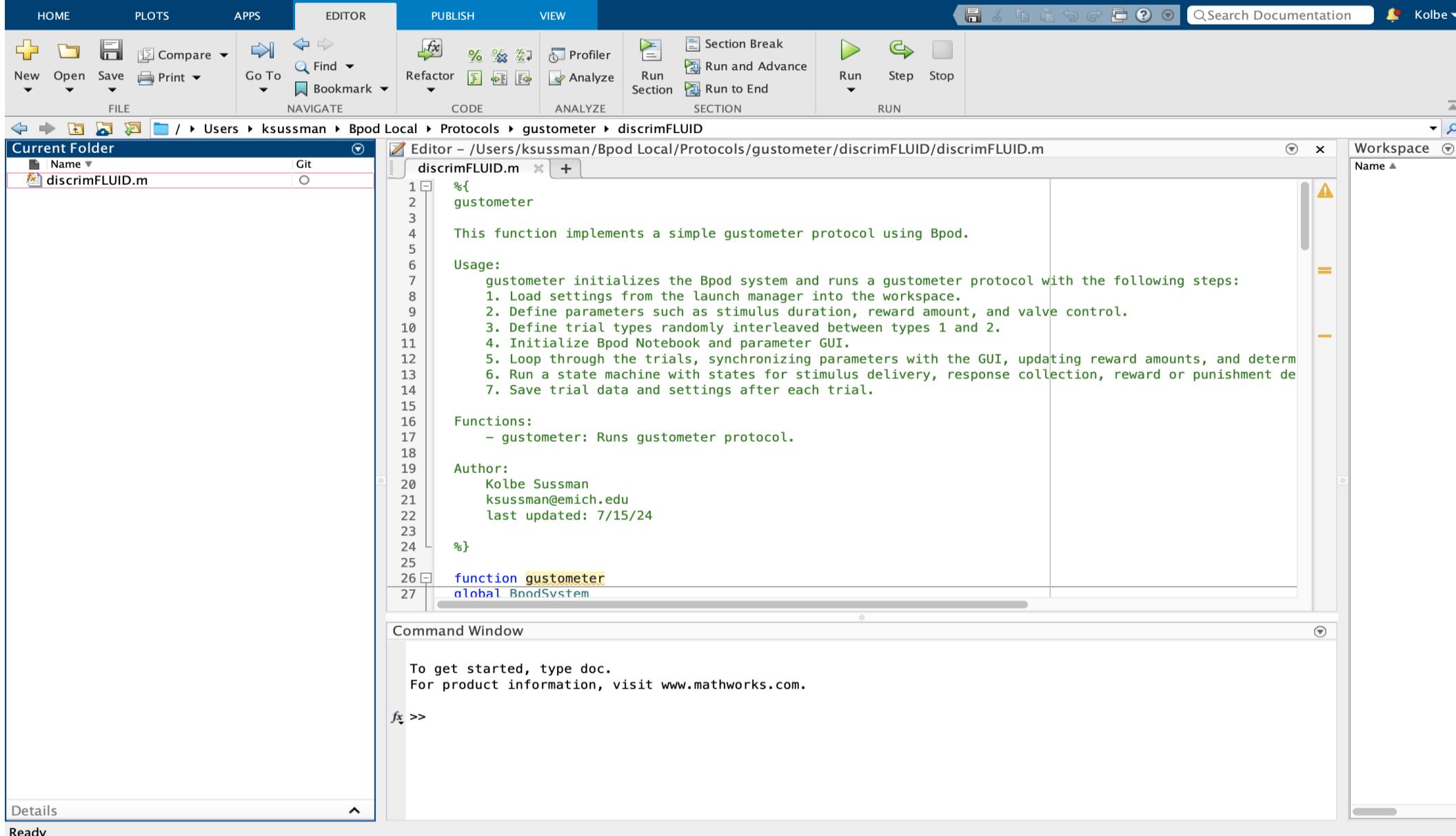
You can open the output .csv file (which will always be named “output_[date].csv” in MATLAB (top) by double clicking on it in the ‘Current Folder’ panel (last slide 1). You can also open it in excel (bottom), R, python, or any other program that can import a CSV.

Understanding Bpod Protocol Scripts

Required:

- Matlab ([Download here](#), EMU Students get a free download on one personal device)
- Bpod Software ([Download Here](#); the first time you download it it will ask you to setup, just select all default options)
- Gustometer Protocols ([Download Here](#))

Note: I will be going through `DiscriminationFLUID.m` for this example because it has semi-complicated logic and if you understand this script you should be able to understand the others and write your own.



Open the MATLAB script file ('.m' files). The beginning of the script contains basic documentation defining the script's function. If you update a protocol, change the 'last updated:' line and add a line that says 'collaborators: [Your Name]'. For the remainder of this tutorial, I will be cropping the screenshot to only include the editor window.

The screenshot shows a MATLAB Editor window with the following code:

```
Editor - /Users/ksussman/Bpod Local/Protocols/gustometer/discrimFLUID/discrimFLUID.m
discrimFLUID.m + ⚡
```

```
25
26 function gustometer
27     global BpodSystem
28     MaxTrials = 1000;
29
30     %% Define parameters
31     S = BpodSystem.ProtocolSettings; % Load settings chosen in launch manager into current workspace as a
32     if isempty(fieldnames(S)) % If settings file was an empty struct, populate struct with default settin
33         S.GUI.StimulusDuration = .1; % in seconds
34         S.GUI.RewardAmount = 5; %ul
35         S.GUI.OpenValveOne = 'OpenValve(1)';
36         S.GUIMeta.OpenValveOne.Style = 'pushbutton';
37         S.GUI.OpenValveTwo = 'OpenValve(3)';
38         S.GUIMeta.OpenValveTwo.Style = 'pushbutton';
39     end
40
41     %% Define trials
42     TrialTypes = round(rand(1,MaxTrials)) + 1; % Randomly interleaved trial types 1 and 2
43     BpodSystem.Data.TrialTypes = TrialTypes; % The trial type of each trial completed will be added here.
44
45     %% Initialize plots
46     BpodNotebook('init'); % Bpod Notebook (to record text notes about the session or individual trials)
47     BpodParameterGUI('init', S); % Initialize parameter GUI plugin
48
49     %% Main trial loop
50     for currentTrial = 1:MaxTrials
51         S = BpodParameterGUI('sync', S); % Sync parameters with BpodParameterGUI plugin
```

First line of code is always
'function [protocol]'.
Ensure the function name, .m
file name, and folder name are
all the same.

I leave the Define
parameters, Define
trials, and Initialize
plots the same for every
protocol.

Editor - /Users/ksussman/Bpod Local/Protocols/gustometer/discrimFLUID/discrimFLUID.m

discrimFLUID.m

```
48
49 %% Main trial loop
50 for currentTrial = 1:MaxTrials
51     S = BpodParameterGUI('sync', S); % Sync parameters with BpodParameterGUI plugin
52     R = GetValveTimes(S.GUI.RewardAmount, [1 3]); % Update reward amounts
53
54 % Determine trial-specific state matrix fields based on trial type
55 switch TrialTypes(currentTrial)
56     case 1
57         LeftAction = 'Punish'; RightAction = 'Reward';
58         StimulusOutputActions = {'ValveModule1', 2}; % Action to open valve for trial type 1
59         RewardOutputActions = {'ValveModule1', 1}; % Action to deliver reward
60
61     case 2
62         LeftAction = 'Reward'; RightAction = 'Punish';
63         StimulusOutputActions = {'ValveModule1', 3}; % Action to open valve for trial type 2
64         RewardOutputActions = {'ValveModule1', 8}; % Action to deliver reward
65
66 end
67
68 sma = NewStateMachine(); % Initialize new state machine description
69
70 % Define states for stimulus delivery and response collection
71 sma = AddState(sma, 'Name', 'WaitForPoke', ...
72                 'Timer', 0, ...
73                 'StateChangeConditions', {'Port2In', 'DeliverStimulus'}, ...
74                 'OutputActions', {});
```

For tasks where you are delivering a stimulus in the middle and a reward or punishment on the left/right sides, you will need to define trial types and parameters.

In this example we are defining trial type 1 (stimulus from valve 2, reward on right side from valve 1, and punishment on left side) and trial type 2 (stimulus from valve 3, reward on left side from valve 8, and punishment on right side)

Editor - /Users/ksussman/Bpod Local/Protocols/gustometer/discrimFLUID/discrimFLUID.m

```

discrimFLUID.m + 67
68
69     sma = NewStateMachine(); % Initialize new state machine description
70
71     % Define states for stimulus delivery and response collection
72     sma = AddState(sma, 'Name', 'WaitForPoke', ...
73         'Timer', 0, ...
74         'StateChangeConditions', {'Port2In', 'DeliverStimulus'}, ...
75         'OutputActions', {});
76
76     sma = AddState(sma, 'Name', 'DeliverStimulus', ...
77         'Timer', S.GUI.StimulusDuration,... % Duration of stimulus presentation
78         'StateChangeConditions', {'Tup', 'CloseStimulusValve', 'Port2Out', 'CloseStimulusValve'}, ...
79         'OutputActions', StimulusOutputActions);
80
81     sma = AddState(sma, 'Name', 'CloseStimulusValve', ...
82         'Timer', 0.1,... % Immediately close the stimulus valve
83         'StateChangeConditions', {'Tup', 'WaitForResponse'}, ...
84         'OutputActions', StimulusOutputActions); % Action to close the valve
85
86     sma = AddState(sma, 'Name', 'WaitForResponse', ...
87         'Timer', 0, ...
88         'StateChangeConditions', {'Port1In', LeftAction, 'Port3In', RightAction}, ...
89         'OutputActions', {});
90
91     % Define states for reward delivery or punishment
92     sma = AddState(sma, 'Name', 'Reward', ...
93         'Timer', S.GUI.StimulusDuration,... % Duration of reward delivery
94         'StateChangeConditions', {'Tup', 'CloseRewardValve'}, ...
95         'OutputActions', RewardOutputActions);
96
97     sma = AddState(sma, 'Name', 'CloseRewardValve', ...
98         'Timer', 0.1,... % Immediately close the reward valve
99         'StateChangeConditions', {'Tup', 'exit'}, ...
100        'OutputActions', RewardOutputActions); % Action to close the valve
101
102    sma = AddState(sma, 'Name', 'Punish', ...
103        'Timer', 3,... % Duration of punishment
104        'StateChangeConditions', {'Tup', 'exit'}, ...
105        'OutputActions', {});
106
107    SendStateMachine(sma); % Send state machine to Bpod
108
109

```

Editor - /Users/ksussman/Bpod Local/Protocols/gustometer/discrimFLUID/discrimFLUID.m

```

discrimFLUID.m + 67
68
69     sma = NewStateMachine(); % Initialize new state machine description
70
71     % Define states for stimulus delivery and response collection
72     sma = AddState(sma, 'Name', 'WaitForPoke', ...
73         'Timer', 0, ...
74         'StateChangeConditions', {'Port2In', 'WaitForResponse'}, ...
75         'OutputActions', StimulusOutputActions); % Action to close the valve
76
76     sma = AddState(sma, 'Name', 'WaitForResponse', ...
77         'Timer', 0, ...
78         'StateChangeConditions', {'Port1In', LeftAction, 'Port3In', RightAction}, ...
79         'OutputActions', {});
80
81     % Define states for reward delivery or punishment
82     sma = AddState(sma, 'Name', 'Reward', ...
83         'Timer', S.GUI.StimulusDuration,... % Duration of reward delivery
84         'StateChangeConditions', {'Tup', 'CloseRewardValve'}, ...
85         'OutputActions', RewardOutputActions);
86
87     sma = AddState(sma, 'Name', 'CloseRewardValve', ...
88         'Timer', 0.1,... % Immediately close the reward valve
89         'StateChangeConditions', {'Tup', 'exit'}, ...
90         'OutputActions', RewardOutputActions); % Action to close the valve
91
92     sma = AddState(sma, 'Name', 'Punish', ...
93         'Timer', 3,... % Duration of punishment
94         'StateChangeConditions', {'Tup', 'exit'}, ...
95         'OutputActions', {});
96
97     SendStateMachine(sma); % Send state machine to Bpod
98
99

```

Next we will define our states. This is the “beef” of our protocol. Initialize the state machine first. Then add all desired states. Defining each state follows a specific format:

```

sma = AddState(sma, 'Name', '[name]', ...
    'Timer', [time in seconds],...
    'StateChangeConditions', {[condition], [next state]},...
    'OutputActions', {[box action]});

```

Important notes about states:

- Each state should have a DESCRIPTIVE name
- Each state should have a timer. A timer set to 0 means that it will hold the state until a state change condition is met.
- Each state should have at least one state change condition! A given state can have more than one condition and they must be defined in the format of “[condition], [next state], [condition2], [next state2]”. If the state changes when a timer is up the condition should be set to ‘tup’. If the timer is set to 0, the condition **must** be a behavior of the mouse (i.e., ‘Port2In’). The name of the next state must be specified. If you want the trial to end (i.e., you’ve delivered a punishment or reward) set the next state to ‘>exit’ (you do not need to define the ‘>exit’ state, the Bpod knows what this means).
- Each state must contain an output action statement BUT you don’t need to use one. Output action must be a behavior of the box (i.e., “{‘Valve’, 1}” or “{‘LED’, 3}”). This action will be held for the duration of the state*. If you don’t want the box to do anything (i.e. you want it to wait for a poke) set the output action to “{}”.

* Valves do not close at the end of a timer. You must add a Valve close state where the timer is short (i.e., 0.1 seconds) and the output action is the same as the valve open state. All valve open states MUST lead to valve close state and then the valve close state can lead to the next desired state.

Editor - /Users/ksussman/Bpod Local/Protocols/gustometer/discrimFLUID/discrimFLUID.m

discrimFLUID.m

```
106
107     SendStateMachine(sma); % Send state machine to Bpod
108
109     RawEvents = RunStateMachine; % Run the state machine
110
111     % If trial data was returned, record trial events and settings
112     if ~isempty(fieldnames(RawEvents))
113         BpodSystem.Data = AddTrialEvents(BpodSystem.Data, RawEvents); % Computes trial events from raw dat
114         BpodSystem.Data = BpodNotebook('sync', BpodSystem.Data); % Sync with Bpod notebook plugin
115         BpodSystem.Data.TrialSettings(currentTrial) = S; % Adds the settings used for the current trial to
116         SaveBpodSessionData; % Saves the field BpodSystem.Data to the current data file
117     end
118
119     HandlePauseCondition; % Checks to see if the protocol is paused. If so, waits until user resumes.
120
121     if BpodSystem.Status.Used == 0
122         return
123     end
124 end
125
```

I leave all this the same, this is what saves the data

Other useful links:

Required:

- [Hardware](#)
- [Software Download](#)
- Bpod Wiki ([Hardware](#) and [Software API](#))
- [Support Forums](#)