

Jul 22, 2020

OptoPlate Calibration protocol

Edvard Grødem¹, Kieran Sweeney¹, Megan N. McClean¹¹Department of Biomedical Engineering, University of Wisconsin-Madison, Madison, WI USA**1** Works for me dx.doi.org/10.17504/protocols.io.bivmke46

Kieran Sweeney

ABSTRACT

Optogenetic systems use light to precisely control and investigate cellular processes. Until recently, there had been few instruments available for applying controlled light doses to cultures of cells. The optoPlate, a programmable array of 192 LEDs, was developed to meet this need. However, LED performance varies and without calibration there are substantial brightness differences between LEDs on an optoPlate. Here we present a method for calibrating an optoPlate that uses a programmable microscope stage and optical power meter to automatically measure all 192 LEDs of an optoPlate. The resulting brightness measurements are used to calculate calibration values that tune the electrical current supplied to each optoPlate LED to reduce brightness variation in optogenetic experiments.

THIS PROTOCOL ACCOMPANIES THE FOLLOWING PUBLICATION

Grødem, Sweeney & McClean, Automated calibration of optoPlate LEDs, Biotechniques, July 2020, DOI: 10.2144/btn-2020-0077

DOI

dx.doi.org/10.17504/protocols.io.bivmke46

PROTOCOL CITATION

Edvard Grødem, Kieran Sweeney, Megan N. McClean 2020. OptoPlate Calibration protocol. **protocols.io** dx.doi.org/10.17504/protocols.io.bivmke46

MANUSCRIPT CITATION please remember to cite the following publication along with this protocol

Grødem, Sweeney & McClean, Automated calibration of optoPlate LEDs, Biotechniques, July 2020, DOI: 10.2144/btn-2020-0077

KEYWORDS

optogenetics, optoPlate, calibration

LICENSE

This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

CREATED

Jul 22, 2020

LAST MODIFIED

Jul 22, 2020

PROTOCOL INTEGER ID

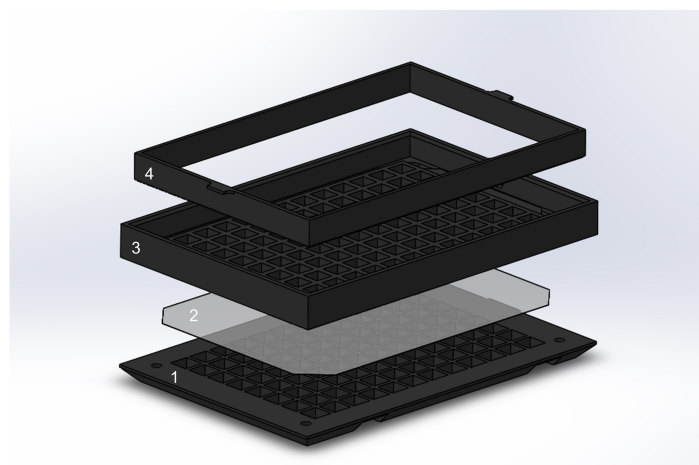
39565

Configuring the optoPlate hardware



Section overview

Figure S1



1. OptoPlateToNunc265300_BottomHalf
2. Diffuser sheets (Rosco, #3008)
3. OptoPlateToNunc265300_TopHalf
4. OptoPlateToNunc265300_Top

We assemble the optoPlate as published by [Bugaj and Lim](#), but with two changes to suit the needs of our lab and allow calibration. First, we adapt the optoPlate to accommodate our 96 well optical-bottom plates (Nunc, #265300) via 3D printed adaptors, which can be found in [our GitHub page](#). By default, these adaptors allow one to mount the 96 well plate on top of the optoPlate, though we also created a secondary adaptor to mount the optoPlate upside-down on top of the 96 well plate. The optoPlate can be equipped with these adaptors as depicted in Figure S1.

- 2 Mount the optoPlateToNunc265300_BottomHalf adaptor to the optoPlate PCB
- 3 Place three diffuser sheets (Rosco, #3008) on top of the optoPlateToNunc265300_BottomHalf adaptor
- 4 Mount the optoPlateToNunc265300_TopHalf adaptor on top of the diffuser sheets. At this stage, a Nunc 96 well optical-bottom plate can be mounted on top of the optoPlate
- 5 Slot the optoPlateToNunc265300_Top adaptor into the optoPlateToNunc265300_TopHalf. The optoPlate can now be mounted upside-down on top of a Nunc 96 well optical-bottom plate.

Configuring the optoPlate firmware



Section overview

We developed new firmware for the optoPlate so the brightness of each LED is controlled by both a calibration value and an input value as follows:

$$LEDintensity \propto (inputvalues)/255 * (calibrationvalues)/255$$

The input value is supplied by a user to control LED brightness over the course of an experiment. The calibration value must be set only once and accounts for brightness differences between LEDs, so that two LEDs set to the same input value will have the same brightness. The optoPlate firmware was built using PlatformIO and the onboard Arduino and it allows users to design LED light programs and flash them to the optoPlate using MATLAB.

The firmware used in this protocol is available on [our GitHub page](#). Instructions for installing the firmware are included in the README file, as are instructions for creating light programs for experiments and flashing them to the optoPlate.

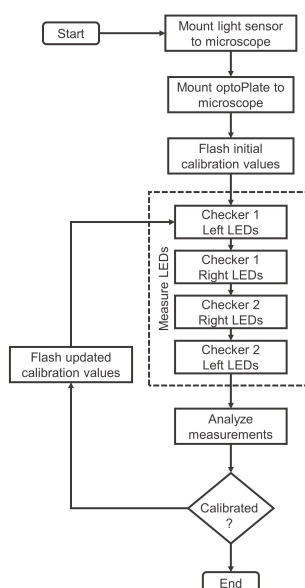
Calibrating an optoPlate with two LEDs

7



Section overview

Figure S2

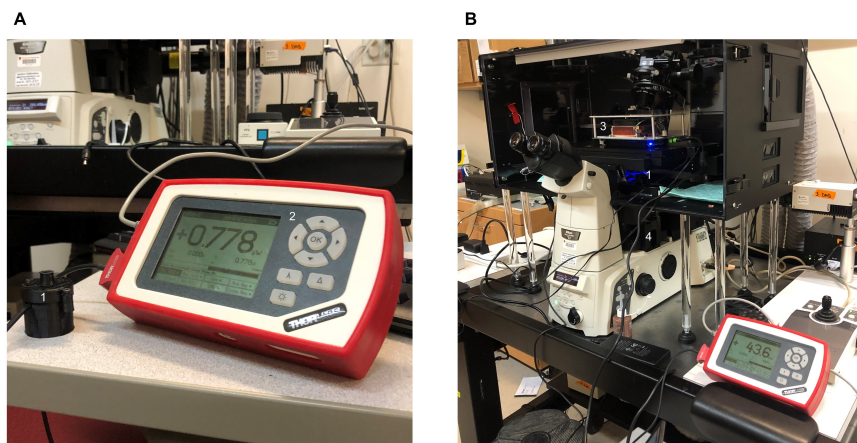


Our calibration method (depicted in Figure S2) uses the programmable stage of a microscope to automatically move an optical power sensor over each optoPlate LED in a set order to acquire a timeseries of irradiance measurements from which the brightness of each LED can be determined and used to calculate calibration values. To ensure that each LED creates a distinct peak in the irradiance signal, we do not measure all the LEDs in one pass. Instead we measure all LEDs over the course of four passes in which we raster the optical power meter across the optoPlate such that every other LED encountered is inactive, leading to an irradiance signal in which measurements from inactive LEDs create clear troughs between

measurements of bright LEDs. We do this by setting the optoPlate to deliver a checkerboard illumination pattern, such that no two illuminated wells are directly above, below, or beside each other in a given pass. We measure the remaining wells by setting the optoPlate to deliver the inverse checkerboard illumination pattern. Because each well of the optoPlate has two LEDs, we repeat this process twice, once for the left LED and once for the right LED. Once all 192 LEDs are measured, we analyze the irradiance measurements using a MATLAB script that determines the brightness of each LED and calculates a calibration value so that the brightest LEDs are dimmed to match the dimmest LED. We then flash the calibration values to the optoPlate and repeat the whole process until the coefficient of variation of the LED irradiance measurements falls below a set threshold (we use 1%).

8

Figure S3



1. Thorlabs S120VC optical power meter mounted between the upper and lower optical power sensor adapters. Shown on its own (A) and mounted to the objective turret of a microscope (B)
2. Thorlabs PM100D records irradiance measurements
3. optoPlate mounted to a Nunc 96 well plate optical bottom plate atop a programmable microscope stage via the optoPlate-to-Nunc adapter and with the secondary optoPlate-to-Nunc adapter (to mount the optoPlate upside-down)
4. Microscope with programmable stage used to raster the optical power sensor across the optoPlate LEDs

Mount a refocusing unit (described by [Grünwald et al](#)) to the active objective socket of a Nikon TiE inverted microscope. Then (as shown in Figure S3A) fasten the optical power sensor (Thorlabs S120VC) between the upper and lower optical power sensor adaptors (MicroscopeAdaptorLowerPart and MicroscopeAdaptorUpperPart, respectively) and mount the resulting assembly on the refocusing unit. Then connect the Thorlabs power sensor to a Thorlabs power meter (PM100D), which should be connected to a computer running Thorlabs Optical Power Monitor Version 1.1 software

- 9 Mount a Nunc 96 well optical-bottom plate to the microscope stage (Figure S3B). Then mount the optoPlate upside-down on top of the 96 well plate via the optoPlateToNunc265300_Top adaptor. The optoPlate should be positioned so that well A1 of the optoPlate is positioned over well A12 of the 96 well plate. Plug in the optoPlate and connect it to a computer via the USB cable. To finish preparing the microscope for calibration, adjust its focus knob to raise the optical power sensor until it nearly touches the 96 well plate. Note the z-position of the stage and use it for all future measurements (we always use a z-position of 4000 μm). This approach allows consistent micron level precision when setting the distance between the optical power sensor and optoPlate.
- 10 Flash the initial set of calibration values to the optoPlate by running the MATLAB script 'flash_calibration.m' and selecting the file 'cal_round_0' at the prompt. This sets the initial calibration value of each LED to 255
- 11 Making sure that the optical power meter is set to measure the appropriate wavelength for each set of LEDs, measure the brightness of the LEDs by using the programmable microscope stage to raster the optical power sensor across the

96 well plate:

- 11.1 Flash the first checkerboard illumination pattern to the optoPlate by running 'FlashExperiment.m' and selecting 'CHECKER01.mat' when prompted. Then unplug the power cable for the right LEDs so that only the left LEDs of every other well are active. Start recording irradiance measurements with the power meter software, then have the microscope stage raster through the 96 well plate, pausing at each well for 3 s. We did this via a NIS Elements JOBS module program. Save these measurements as 'ROUND01_LED01_CHECKER01.csv'.
 - 11.2 Unplug the power cable for the left LEDs and plug in the power cable for the right LEDs, then record LED irradiance measurements as described in step 4a. Save these measurements as 'ROUND01_LED02_CHECKER01.csv'
 - 11.3 Flash the second checkerboard illumination pattern to the optoPlate by running 'flash_experiment.m' and selecting 'CHECKER02.mat' when prompted. Record the LED irradiance measurements for both sets of LEDs as described above and save them as 'ROUND01_LED01_CHECKER02.csv' and 'ROUND01_LED02_CHECKER01CHECKER02.csv'
- 12 With all LEDs now measured, run the MATLAB script 'optoPlate_calibration.m' and select the saved '.csv' files at the prompt. Note that this script uses the GRAMM toolbox for plotting, which can be found, along with installation instructions, at github.com/piermorel/gramm. The script will automatically identify each LED from the irradiance measurements and calculate calibration values, which are saved as 'cal_round_1.mat'. It should be noted that this script assumes the wells are measured in a set order in which 1) the microscope rasters through a 96 well plate mounted on its programmable stage from A1-A12 to B12-B1 and so on and 2) the optoPlate is flipped when mounted upside-down on the 96 well plate so that these measurements actually correspond to optoPlate wells A12-A1 to B1-B12 etc. Flashing the optoPlate with CHECKER01.mat sets the checkerboard pattern in which the LEDs of optoPlate well A01 are active, while CHECKER02.mat sets the checkerboard pattern in which the LEDs of optoPlate well A01 are inactive. By default, the calibration script assumes an optoPlate configuration with two LEDs of the same color per well. If this is not the case, set the variable 'calibrate_LEDs_independently' to true and the script will treat each set of LEDs (LED01, LED02, etc.) as separate groups of different colored LEDs when calculating calibration values.
 - 13 Flash the updated calibration values to the optoPlate. This concludes round one of calibration. The brightness of the LEDs should be much more uniform, though more rounds of calibration may be needed to further reduce brightness differences between LEDs
 - 14 Repeat steps 4 – 6 as needed until the coefficient of variation of the LED irradiance measurements falls below a set threshold (we chose 1%). Each of these repeats constitutes another round of calibration. When saving the 'csv' files, the round number should be included in the filename, for example, the left LED measurements for the first checkerboard illumination pattern of round two should be saved as 'ROUND02_LED01_CHECKER01.csv'. Likewise, when loading the measurements into 'optoPlate_calibration.m' or flashing calibration values to the optoPlate, be sure to use the measurements or calibration values for the current round.

Calibrating an optoPlate with three LEDs

15



We described the calibration of an optoPlate with two LEDs per well. However, an optoPlate can also be configured to have three different colored LEDs per well. In this case, the overall calibration process is similar, but involves six measurement passes—one per LED color for each checkerboard pattern. With three LEDs per well, one must switch between active LED sets by flashing checkerboard patterns for each color rather than reconnecting the power cables. As before, one must set 'calibrate_LEDs_independently' to true and set the optical power meter to measure the correct light wavelength for each set of LEDs.