# Changes to DataMerger

The main changes from DataMerger 1.0 to 2.0 are:

* the use of a different output file naming convention
* the incorporation of (optional) unit data
* the introduction of an additional helper function labchart2datamerger which can be used to adapt PowerLab Chart data into “sampsamp–like” format so that it can also be used in DataMerger. See the labchart2datamerger section below for more details.
* a small utility named fixup to repair parameter files that misbehave when opened in Matlab

File Naming

The output file name now begins with a date-time timestamp, followed by the name of the stimulus, the cell number and the trial number.

For example:

2017-04-04@12\_35\_59-TargetOnly-Cell01-Trial8.mat

2017-04-04@12\_44\_13-OpticFlowWithTarget-Cell01-Trial3.mat

Unit Data

If acquisition has been done using PowerLab Chart, the input file containing the data (created using labchart2datamerger) can optionally contain unit data that tracks the occurrence of spikes from identified units.

DataMerger will look for MATLAB variables named unit\_blockN, where N is the number of the block. These are analogous to data\_blockN and ticktimes\_blockN from sampsamp, and correspond to the same blocks in the data.

After merging, DataMerger will include a variable named Units in the .mat output file. This will be a matrix with N rows, where each row corresponds to a different unit. The row for a particular unit contains a 1 at each sample where a spike for that unit occurred, and 0 everywhere else.

The rest of the output file format is unchanged.

Note that sampsamp itself does not currently supply spike/unit data. However, as the inclusion of unit data is optional, old sampsamp files should still load correctly as before (but will have their output saved to files named according to the new convention).

## labchart2datamerger: formatting Chart data for DataMerger

The labchart2datamerger folder contains a number of functions to help the analyst to format data recorded with PowerLab Chart so that it can be merged with the corresponding FlyFly parameter files using DataMerger. DataMerger currently accepts only the data format that is produced when data is acquired using sampsamp. The functions in the labchart2datamerger folder remove this dependence on sampsamp as an acquisition system, so we can use data acquired with LabChart as well.

There are two main functions in the folder:

* find\_blocks detects the boundaries of data blocks (defined as periods when the photo diode is switched on), using photo diode data (created previously with LabChart).
* labchart2datamerger reads in a file of LabChart formatted data, converts it to data that is suitable for use in DataMerger, and saves the converted data to a file.

## find\_blocks

find\_blocks takes one required argument (the photo diode data array), and five optional arguments. The first four optional arguments are parameters that determine how find\_blocks detects blocks of data from the photo diode data. They are in sequence: ON\_THRESHOLD, UPPER\_VALUE\_THRESHOLD, ON\_DURATION, and OFF\_DURATION. The fifth parameter view determines whether the results of block detection are displayed by calling the separate function plot\_blocks (default value is true).

The way find\_blocks detects blocks is as follows:

It first locates potential blocks as all continuous time periods where the value of the photo diode is greater than ON\_THRESHOLD. For each potential block, it then checks whether there are any values greater than UPPER\_VALUE\_THRESHOLD, which is a kind of “sanity check” – if values exceed this limit, it is likely that there was something wrong with the photo diode recording at this point, and the block is discarded.

Next, the software finds and corrects mistakes in the demarcation of blocks. These mistakes can be of two kinds:

1. We have incorrectly split up a block into two or more smaller blocks because the photo diode value dropped below ON\_THRESHOLD, but did so for only a negligible length of time. To fix this, all gaps between blocks that are shorter than OFF\_DURATION samples in duration are ignored, and the two blocks on either side of the gap are merged together.
2. We have incorrectly identified a sequence as a data block when the photo diode value went above ON\_THRESHOLD, but this occurred for only a negligible length of time. To fix this, all blocks shorter than ON\_DURATION samples in duration are discarded. There is an important exception to this: if there is also a negligible gap occurring after this block, the block is not discarded, but is merged with the next block instead.

By default, find\_blocks will use the values 1.5e-3 and 9e-3 respectively for ON\_THRESHOLD and UPPER\_VALUE\_THRESHOLD, and 20 samples each for ON\_DURATION and OFF\_DURATION.

labchart2datamerger

labchart2datamerger takes two required arguments (input file name, output file name) and the four optional parameters described above for find\_blocks, which are passed straight through to find\_blocks.

labchart2datamerger loads an input file which should contain variables named Date, Photodiode, RawData and Time, and optionally variables named Unit1, Unit2, etc.

It calls find\_blocks, then uses the detected block boundaries to package the raw data, photo diode data and unit data into the correct format for DataMerger. It does not call plot\_blocks, so that results are not displayed on the screen.

Finally, the converted data is saved to the output file that was specified.

It is recommended that analysts run find\_blocks on its own first, with the view parameter set to true, in order to determine appropriate parameters for their data set. By default, this will also call plot\_blocks, which will display the block boundaries on the screen. Once the correct parameters have been found, they can then be passed in to labchart2datamerger (if the default values for the parameters turn out to be sufficient, then only the input and output file names need to be passed in).

fixup

Some parameter files created with previous versions of FlyFly exhibited the behaviour that their associated GUI handles were saved inside the parameter file; as a consequence, double-clicking on them in Matlab loaded the FlyFly GUI that they were created with, which then proved difficult or impossible to close in Matlab. To retroactively go back and delete the offending GUI handles from old parameter files, DataMerger2.0 features a utility imaginatively named fixup, residing in the fixup folder.

fixup operates on a whole folder at a time. When run, it prompts the user to navigate to the folder, then creates a subfolder named fixed, and populates it with “fixed” versions of the parameter files it finds in the folder. When using fixup, please check that the fixed parameter files contain the information you want before deleting the original files!