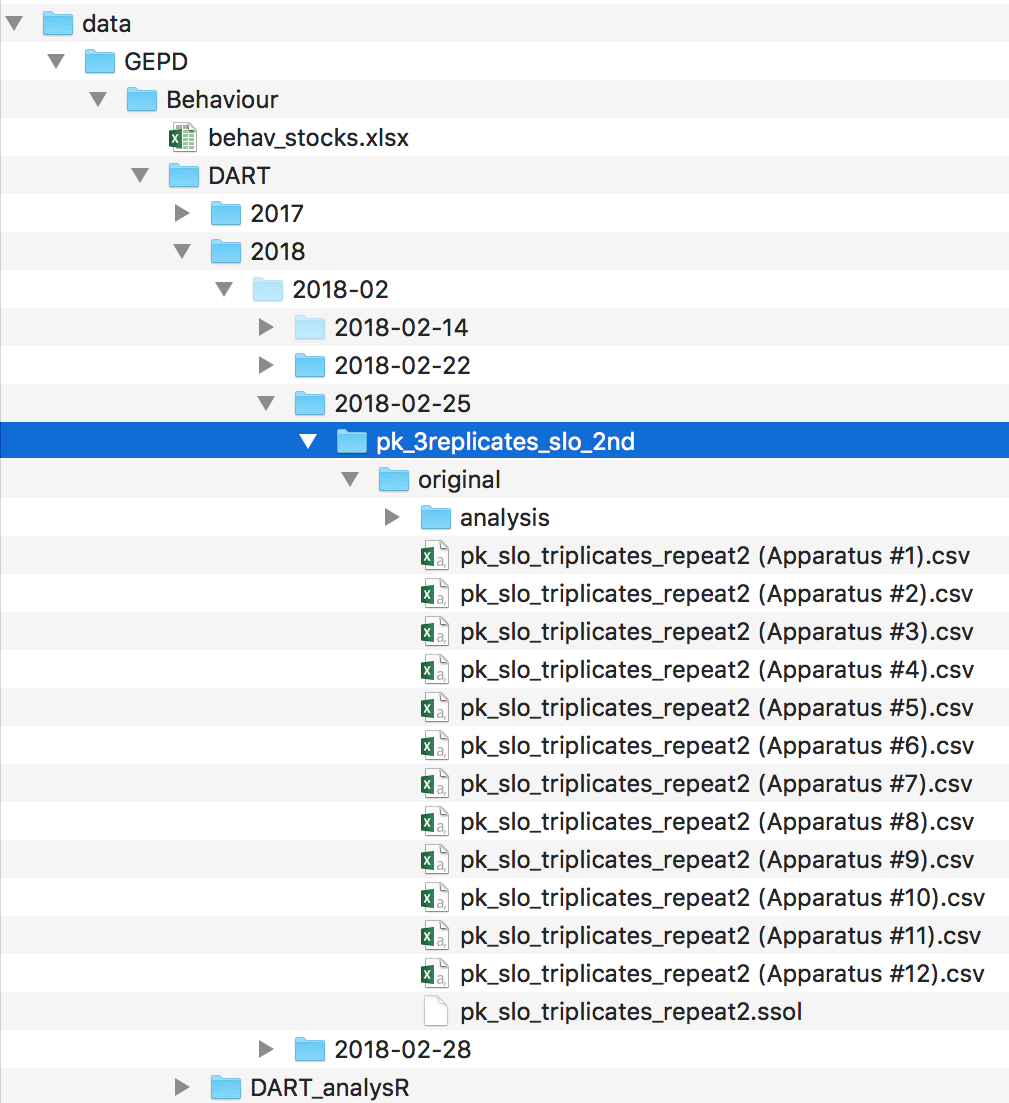
Analysis of DART behavioural data using the DART\_analysR package

Patrick Krätschmer

DART\_analysR takes the raw experimental output from the DART (including y-axis values), and computes the distance travelled by each fly within an optional bin width of time. It takes some time to set up (described below), but the actual script calculates the output within about 1 second per apparatus. Below, I describe briefly what to do in order to use DART\_analysR.

1. Retrieve experimental output files from DART as .csv files (one file per apparatus), **including y-axis data**.

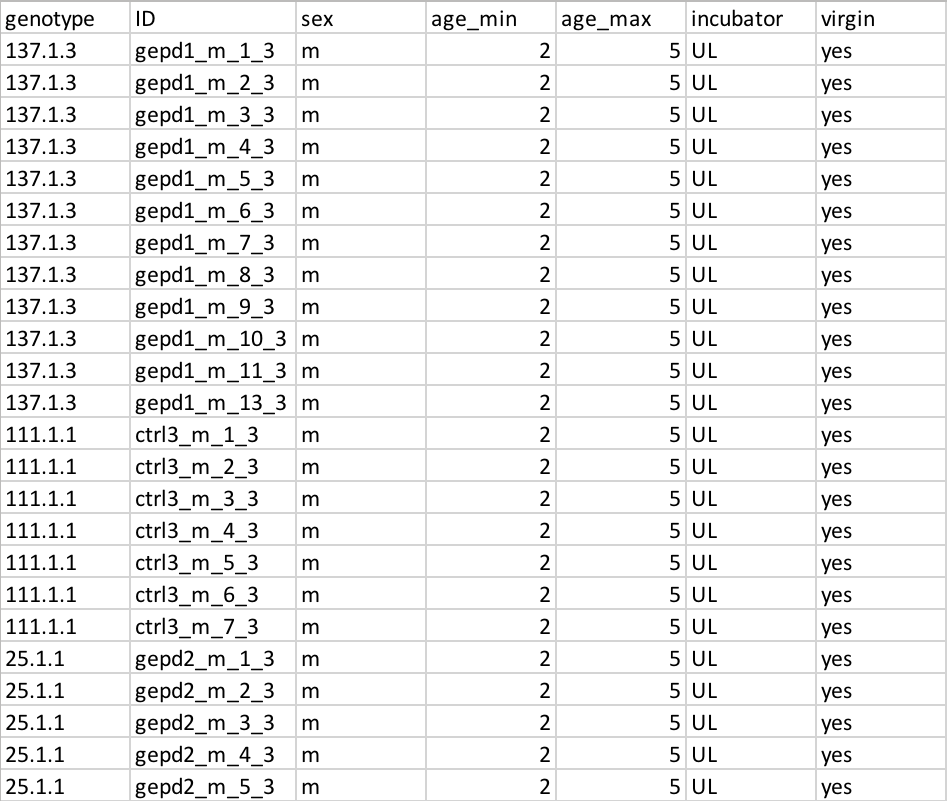
2. **Important**: Your directories should be organised similar to the structure shown in Figure 1. While it doesn't have to be precisely the way shown here, it would be good practice to have at least a systematic organisation, because DART\_analysR uses paths as input.



**Figure 1: Path structure for DART\_analysR usage.** The directory 'pk\_3replicates\_slo\_2nd' is the parent directory. It contains the subdirectory 'original', which contains a subdirectory called 'analysis' and all the DART experimental output files (.csv). The subdirectory 'analysis' contains a .csv file with all the meta-data needed for the DART experiment (described later). Note that 'original' also contains the DART experimental solution .ssol file in this example, but this is not required. The directory 'DART\_analysR' should be located some directories up of 'pk\_3replicates\_slo\_2nd'. In this case, it is in the parent directory 'DART', which contains all DART experiments sorted by date. Note that the names in this example are specific to my own experiments and would be different for you.

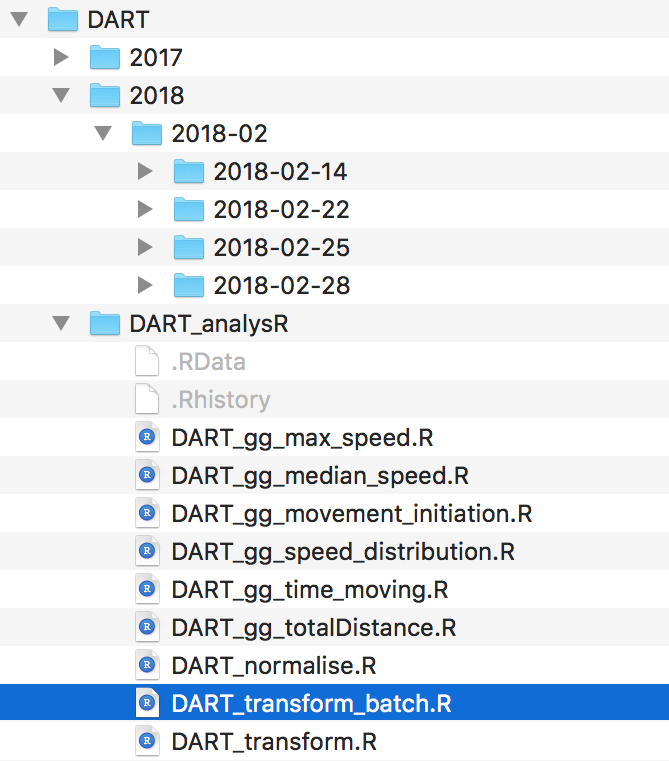
3. Create a .csv file containing all the meta-data needed for the experiment. This file will be in the 'analysis' directory as shown in Figure 1. Figure 2 shows an example of a meta-data file. **IMPORTANT**: Each row in this file corresponds to one fly from your DART experiment. The order is as if all your DART apparati were stacked vertically on top of one another, starting with fly 1 of apparatus 1 and ending with the last fly of the last apparatus. Make sure to account for missing flies, e.g. untracked or dead flies, because otherwise this would cause a frameshift in the meta-data file. **IMPORTANT**: Make sure to always assign the same ID's to genotypes, so you can pool the data later on. The format of the ID's must be followed precisely: <someName>\_<someIdentifier>\_<sampleNumber>\_<replicateNumber>

It is crucial that the string <someName>\_<someIdentifier> has the exact same number of characters throughout all your experiments. Ask me for more details.



**Figure 2: Example meta-data file needed to use DART\_analysR.** The first 2 columns are required, the rest are optional, but it is a good idea to specify as many columns as possible, as this file is a good opportunity to keep track of experimental parameters. **IMPORTANT**: The values in column 2 ('ID') up to the second underscore must have the exact same number of characters. This is due to how DART\_analysR later on parses these ID's to obtain the underlying genotype, e.g. ctrl3\_m\_1\_3, ctrl3\_m\_2\_3, ..., ctrl3\_m\_7\_3 all become ctrl3\_m.

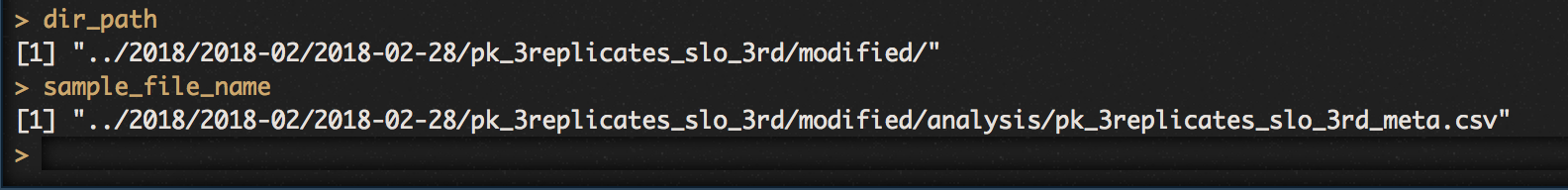
4. Start RStudio by double-clicking on the R script 'DART\_transform\_batch.R', as shown in Figure 3. In RStudio, source this function by typing '> source("DART\_transform\_batch.R")' in the console.



**Figure 3: Start RStudio by double-clicking on 'DART\_transform\_batch.R'.** If RStudio is not the default programme to open R scripts with, change this for all .R files for future use.

5. Before running DART\_transform\_batch(), you need to define two variables (Figure 4):

1. sample\_dir
   1. This needs to be the path to the directory that contains the DART apparatus .csv files. You can get your working directory by calling '>getwd()'. The path to the DART apparatus .csv files must be the **relative** path from your current working directory. Save this path in the variable 'sample\_dir', and don't forget to use double quotes when assigning string variables.
2. sample\_name\_file
   1. This needs to be the path to your DART meta-data .csv file, which should be located in the directory 'analysis' (Figure 1). It should also include the actual meta-data file name.



**Figure 4: Two variables need to be assigned before running DART\_analysR.** 'dir\_path' contains the relative path to the experimental output files (.csv), 'sample\_file\_name' contains the relative path to the meta-data file, **including the file name**.

6. Run DART\_transform\_batch() with the parameters desired. Further explanation of the parameters is inside the R script. Save the output of this function to a variable and continue the analysis with DART\_gg\_\*() plotting functions. If you'd rather just get the output into a csv file, and continue your analysis outside RStudio, you can set the argument 'csv' to TRUE when calling DART\_transform\_batch. This will write your data to a .csv file. Note that this file will be very large (depending on the number of flies). **IMPORTANT**: The output data is 'untidy', i.e. there is one fly per column rather than a single column with the variable 'fly'; the latter would be regarded as 'tidy' but it seems to be easier to work with outside R when the data is organised not in a 'tidy' way.

7. The output of DART\_transform\_batch() is verbose. For example, it will tell you, based on parameters you can set before calling the function, which flies appear to be dead. Hence, after running the function once, you can go back to the DART experimental output and your meta-data, and delete flies that seem to be dead. Thereafter, you can run DART\_transform\_batch() again, only with flies that are defined to have been alive. It is good practice to no modify the 'original' directory's data (Figure 1), so that you can keep the original data. Rather, copy the whole 'original' directory and rename it to 'modified', for example. Then you can delete the dead flies in the 'modified' directory and run DART\_transfrom\_batch() again.