



HEBB'S Lab

Human Enhancement Brain and Body
Systems Science Laboratory

Quick guide for pre-processing experiment data for motor tasks

Contents

1. Basics in behavioral data analysis (for motor tasks in this lab)
2. Organization of directory & Data
3. Procedure (quick manual)

1. Basics in behavioral data analysis (for motor tasks in this lab)

Behavioral experiment & analysis basic steps

1) Data collection (i.e., run experiments)

- Data are stored as text files: row=time, column= data type (position, velocity, etc.)

2) Preprocessing

This slide is about this

- Convert raw files into “midfiles”
 - Facilitate data handling (labeling data, organizing, etc.)
 - Initial processing
 - “zero-ing”, alignment, etc.
 - Smoothing, filtering
 - Data-size reduction (down-sampling)
 - Usually, 1000Hz is too high for our motor tasks (200Hz is typically good enough)

3) Analysis

- Data plot
- Stats
- etc.

2. Organization of directory & Data

R script packet for pre-processing is available on Github

The screenshot shows a GitHub repository page. At the top, there's a dark header with the GitHub logo, a search bar, and navigation links for Pull requests, Issues, Marketplace, and Explore. Below the header, the repository name 'hebbs-manipulandum / manipulandum_initial_process' is displayed, along with a 'Public' badge. To the right of the repository name are buttons for Edit Pins, Unwatch (with a count of 1), and Fork (with a count of 0). The main navigation bar below the repository name includes links for Code (which is underlined in red), Issues (1), Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.

The 'Code' section shows the 'master' branch with 5 branches and 0 tags. It lists four commits from the 'sugiyama-hebbs' user:

- function: Organize files and small edits on the "main" script (14 months ago)
- script: Temporal data alignment has been added. Also, fixed an important b... (13 months ago)
- .gitignore: another test commit (13 months ago)
- README.md: Create README.md (14 months ago)

Below the code area, there's a preview of the 'README.md' file content:

```
manipulandum_initial_process

This is a set of R scripts for initial processing on manipulandum "raw" data files, creating midfiles that you can use for your own analyses. For a moment, this is tuned for my experiment task (i.e., visuomotor rotation task), which will be edited to accommodate other types of tasks (e.g., FF adaptaiton task).
```

On the right side of the page, there are sections for About, Releases, Packages, and Languages. The 'About' section notes 'No description, website, or topics provided.' The 'Releases' section says 'No releases published' and 'Create a new release'. The 'Packages' section indicates 'No packages published' and 'Publish your first package'. The 'Languages' section shows a single entry for R at 100.0%.

Directory organization

- (main directory)
 - xxx.Rproj – Rproject file (you open Rstudio)
 - data
 - rawdata: this is where you put text data file
 - processed: output directory
 - figure
 - function
 - script
 - “main” scripts are located here
 - sub_script: Main scripts call and use these

* My style is largely influenced by this tutorial: so check Lesson 1 and 2 (and more as needed)

<https://pagepiccinini.com/r-course/>

For each block, an experiment script makes 4 types of text file

One file per block

- `data_load`: “sequence” file used for the block
- `data_param`: key parameter of experiment task

One file per trial

- `data_point`: some important single values per trial
(e.g., xy-position at crossing, endpoint-angle)
- `data_kin`: kinematic data at each recording

```
-3.424030000000000129e-01 3
0.0000000000000000e+00 0.
0.0000000000000000e+00 0.
2.0000000000000000e+00 6.
7.240000000000342e-04 -5
0.0000000000000000e+00 0.
-3.424030000000000129e-01 3
0.0000000000000000e+00 0.
0.0000000000000000e+00 0.
3.0000000000000000e+00 6.
7.240000000000342e-04 -5
0.0000000000000000e+00 0.
-3.359630000000000116e-01 3
0.0000000000000000e+00 0.
0.0000000000000000e+00 0.
4.0000000000000000e+00 6.
7.240000000000342e-04 -5
0.0000000000000000e+00 0.
-3.359630000000000116e-01 3
0.0000000000000000e+00 0.
0.0000000000000000e+00 0.
5.0000000000000000e+00 6.
7.240000000000342e-04 -5
0.0000000000000000e+00 0.
-3.359630000000000116e-01 3
0.0000000000000000e+00 0.
0.0000000000000000e+00 0.
6.0000000000000000e+00 6.
7.240000000000342e-04 -5
0.0000000000000000e+00 0.
-3.19772999999999739e-01 3
0.0000000000000000e+00 0.
0.0000000000000000e+00 0.
7.0000000000000000e+00 6.
7.240000000000342e-04 -5
```

data_
Plain

kinematic data

- row: time point (each recording)
- column: data type (x-position, y-position, etc.) * specified in an experiment script, and also labeled in pre-processing

```

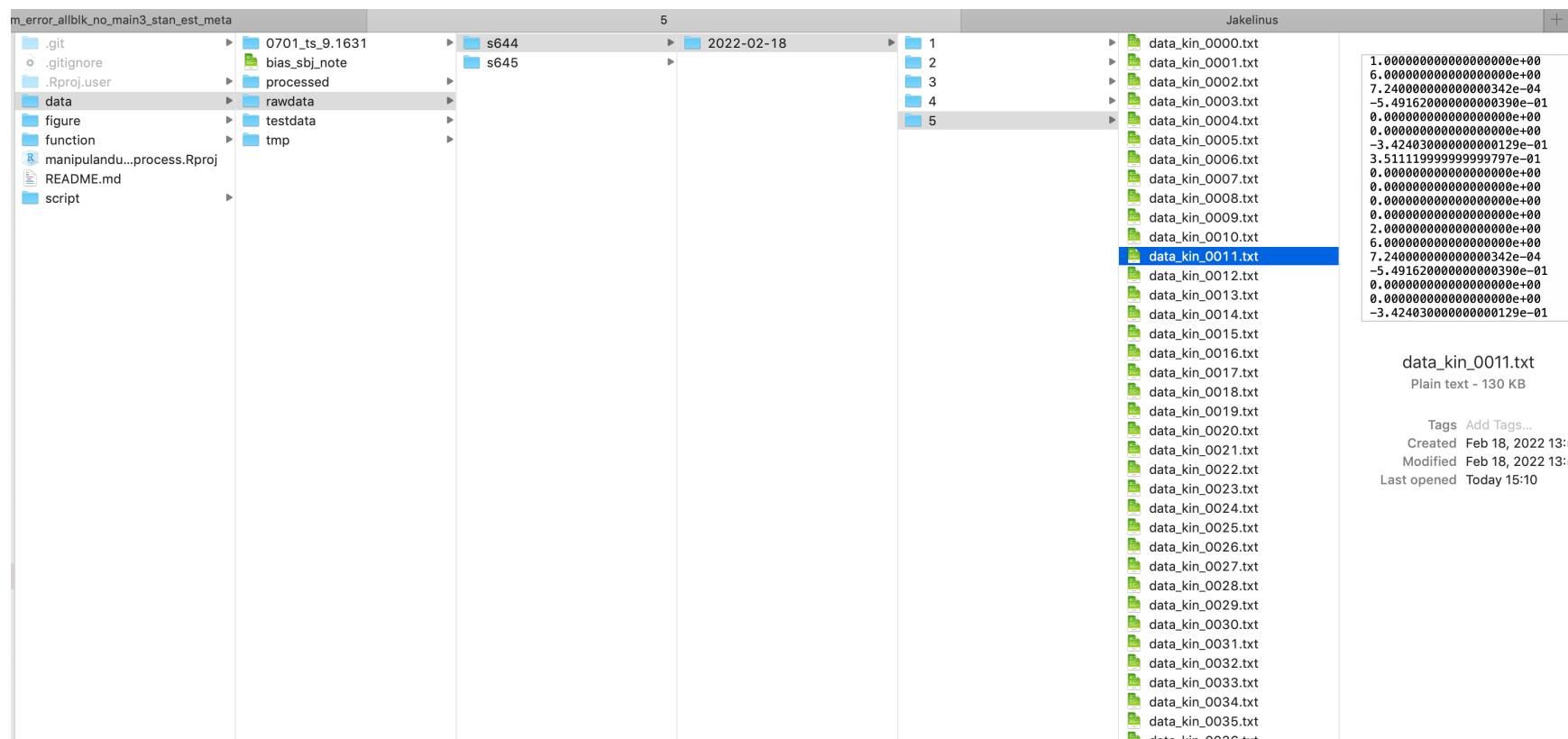
1 |1.00000000000000e+00 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.42403000000000e-01 3.51111999999999797e-0
2 |2.00000000000000e+00 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.42403000000000e-01 3.51111999999999797e-0
3 |3.00000000000000e+00 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.35963000000000e-01 3.5111199999999942e-0
4 |4.00000000000000e+00 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.35963000000000e-01 3.5111199999999942e-0
5 |5.00000000000000e+00 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.35963000000000e-01 3.5111199999999942e-0
6 |6.00000000000000e+00 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.19772999999999739e-01 3.51358999999999769e-0
7 |7.00000000000000e+00 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.28490000000000e-01 3.77332999999999740e-0
8 |8.00000000000000e+00 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.35480999999999737e-01 3.815109999999999890e-0
9 |9.00000000000000e+00 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.36353000000000e-01 3.81500999999999790e-0
10 |1.00000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.35909999999999864e-01 4.09335999999999776e-0
11 |1.10000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.35708999999999796e-01 4.22040000000000e-026e-0
12 |1.20000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.35699000000000e-01 4.22689999999999902e-0
13 |1.30000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.34826999999999857e-01 4.22680000000000002e-0
14 |1.40000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.34826999999999857e-01 4.22680000000000002e-0
15 |1.50000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.34826999999999857e-01 4.22680000000000002e-0
16 |1.60000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.34826999999999857e-01 4.22680000000000002e-0
17 |1.70000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.27768999999999770e-01 4.2275699999999939e-0
18 |1.80000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.18666000000000e-01 4.22856000000000e-0096e-0
19 |1.90000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.18666000000000e-01 4.22856000000000e-0096e-0
20 |2.00000000000000e+01 6.00000000000000e+00 7.24000000000000e-04 -5.49162000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.18666000000000e-01 4.22856000000000e-0096e-0
21 |2.10000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.18715999999999983e-01 4.19713999999999760e-0
22 |2.20000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.187289999999999845e-01 4.188939999999999886e-0
23 |2.30000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.16226000000000000071e-01 4.189209999999999878e-0
24 |2.40000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.164730000000000043e-01 4.033479999999999843e-0
25 |2.50000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.166519999999999890e-01 3.92079000000000108e-0
26 |2.60000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.166519999999999890e-01 3.92079000000000108e-0
27 |2.70000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.16786999999999983e-01 3.835629999999999853e-01 3.835629999999999873e-0
28 |2.80000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.167869999999999853e-01 3.835629999999999853e-01 3.835629999999999873e-0
29 |2.90000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.167869999999999853e-01 3.835629999999999853e-01 3.835629999999999873e-0
30 |3.00000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.2997500000000000183e-01 3.834190000000000098e-0
31 |3.10000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.347930000000000073e-01 3.8336599999999999846e-0
32 |3.20000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.347930000000000073e-01 3.8336599999999999846e-0
33 |3.30000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.351939999999999920e-01 3.918769999999999754e-0
34 |3.40000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.3810600000000000176e-01 3.88593999999999950e-0
35 |3.50000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.380549999999999944e-01 3.918460000000000276e-0
36 |3.60000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.380549999999999944e-01 3.918460000000000276e-0
37 |3.70000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.380279999999999951e-01 3.935159999999999769e-0
38 |3.80000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.380279999999999951e-01 3.935159999999999769e-0
39 |3.90000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.350420000000000065e-01 4.014039999999999830e-0
40 |4.00000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.349449999999999927e-01 4.075150000000000161e-0
41 |4.10000000000000e+01 6.00000000000000e+00 5.95999999999999586e-04 -5.49077000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.349150000000000182e-01 4.094039999999999901e-0
42 |4.20000000000000e+01 6.00000000000000e+00 6.79000000000000e-04 -5.48946000000000e-01 0.00000000000000e+00 0.00000000000000e+00 -3.349150000000000182e-01 4.094039999999999901e-0

```

3. Procedure (quick manual)

1. Put experiment data under data/rawdata

The example shows s644 (subject ID), which participated on Feb 18, 2022 and performed 5 blocks



2. Open Rproject, open “script/process_rawdata_script.R” set appropriate parameters, and run.

- You may run other scripts under the same directory if you know what they are.
- Depending on your experiment program, you may need to modify column labels, number of columns, etc.

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Displays the `process_rawdata_script.R` script. The code sets various parameters like `sub_id`, `add_zero`, and `plot_bias`, and loads packages such as `dplyr`, `purrr`, `signal`, `iemisc`, `gridExtra`, and `grid`. It also sources a file named `process_kin.R`.
- File Browser:** Shows a file tree in the `manipulandum_initial_process` directory. The files listed are:

Name	Size	Modified
miscellaneous	3.2 KB	Sep 1, 2021, 12:51 PM
process_calib_data.R	5.4 KB	Sep 1, 2021, 12:51 PM
process_rawdata_script.R	5.8 KB	Jan 20, 2022, 4:21 PM
process_rawdata_script_multiple.R		
sub_script		
tmp_script.R	316 B	Sep 1, 2021, 12:51 PM
- Console:** Displays the R startup message, license information, natural language support, and a warning about the absence of warranty. It also shows the platform as `x86_64-apple-darwin17.0 (64-bit)`.

(optional) Check process_rawdata_script.R and the subscripts used in it to understand what processing is done, especially on kinematic data

```

21 names(tgt_raw) <- c(tgt_col,"blk_tri")
22
23 param_raw <- data.frame(read.table(fname_param, header = T, row.names = NULL)) # block-wide parameters
24
25 # single-point data
26 point_raw <- map(fname_point,function(fname){
27   tri_filename <- gsub("^.*/","",fname) # remove file path
28   tri_num <- as.numeric(gsub("[^0-9]","",tri_filename))+1 # extract trial number (starting from 0, so add 1)
29   return_df <- data.frame(read.table(fname, header = F, row.names = NULL)) %>%
30     mutate(blk_tri = tri_num) # get block trial number, which is in filename (starting from 0, so add 1)
31 }) %>%
32   reduce(rbind)
33
34 colnames(point_raw) <- c(point_col,"blk_tri")
35
36 # kinematic data
37 task_center_x <- param_raw$center_pos_x
38 task_center_y <- param_raw$center_pos_y
39
40 kin_raw <- map(fname_kin,function(fname){
41
42   tri_filename <- gsub("^.*/","",fname) # remove file path
43   tri_num <- as.numeric(gsub("[^0-9]","",tri_filename))+1 # extract trial number (starting from 0, so add 1)
44   return_df_raw <- data.frame(read.table(fname, header = F, row.names = NULL)) %>%
45
46   colnames(return_df_raw) <- kin_col
47
48   return_df_raw2 <- return_df_raw %>%
49     mutate(x = (x - task_center_x), y = (y - task_center_y)) # zero-ing with respect to the center of task space.
50
51   return_df <- process_kin(return_df_raw2,reduce_hz,reduce_hz_rate) %>%
52     mutate(blk_tri = tri_num)
53
54 }) %>%
55   reduce(rbind)
56
57
+ process_kin <- function(df, reduce_hz, reduce_hz_rate){
+
# pos_data_tri <- subset(pos_data, trial == tri)
+
x <- df$x # raw x position
y <- df$y # raw y position
+
## SG filtering (smoothing & derivatives).
+
# x
sx <- sgolayfilt(x, p = order, n = framelen, m = 0, ts = 1/sample_rate) # smooth
svx <- sgolayfilt(x, p = order, n = framelen, m = 1, ts = 1/sample_rate) # velocity
sax <- sgolayfilt(x, p = order, n = framelen, m = 2, ts = 1/sample_rate) # acceleration
sjx <- sgolayfilt(x, p = order, n = framelen, m = 3, ts = 1/sample_rate) # jerk
+
# y
sy <- sgolayfilt(y, p = order, n = framelen, m = 0, ts = 1/sample_rate) # smooth
svy <- sgolayfilt(y, p = order, n = framelen, m = 1, ts = 1/sample_rate) # velocity
say <- sgolayfilt(y, p = order, n = framelen, m = 2, ts = 1/sample_rate) # acceleration
sjy <- sgolayfilt(y, p = order, n = framelen, m = 3, ts = 1/sample_rate) # jerk
+
vx <- df$vx # robot x-velocity
vy <- df$vy # robot y-velocity
state <- df$state # trial state
+
if (reduce_hz) {
  return_df <- data.frame(tstep = 1:length(x), state, x, y, vx, vy, sx, sy, svx, svy, sax, say, sjx, sjy, fx = df$fs_x, fy = df$fs_y) %>%
    # dplyr::filter(tstep %% red_rate == 1) # reduce sample Hz
    dplyr::filter(tstep %% reduce_hz_rate == 1) # reduce sample Hz
} else{
  return_df <- data.frame(tstep = 1:length(x), state, x, y, vx, vy, sx, sy, svx, svy, sax, say, sjx, sjy, fx = df$fs_x, fy = df$fs_y)
}

```

3. Copy/move output file to your analysis packet

- Midfile is saved as Rdata for each block (or csv if you specify it in process_raw_data_script.R)
- It contains a list with 4 data frames
 - point: “point” values for each trials (e.g., endpoint angle)
 - tgt: sequence file data
 - kin: kinematic data
 - param: block parameter values (e.g., target distance)

Use this for subsequent analysis (plot, stats, etc.)



3. Copy/move output file to your analysis packet

My analysis packets have the same organization of directory, so familiarizing yourself with it helps if you also use them (but feel free to eventually develop your own style of organizing)

