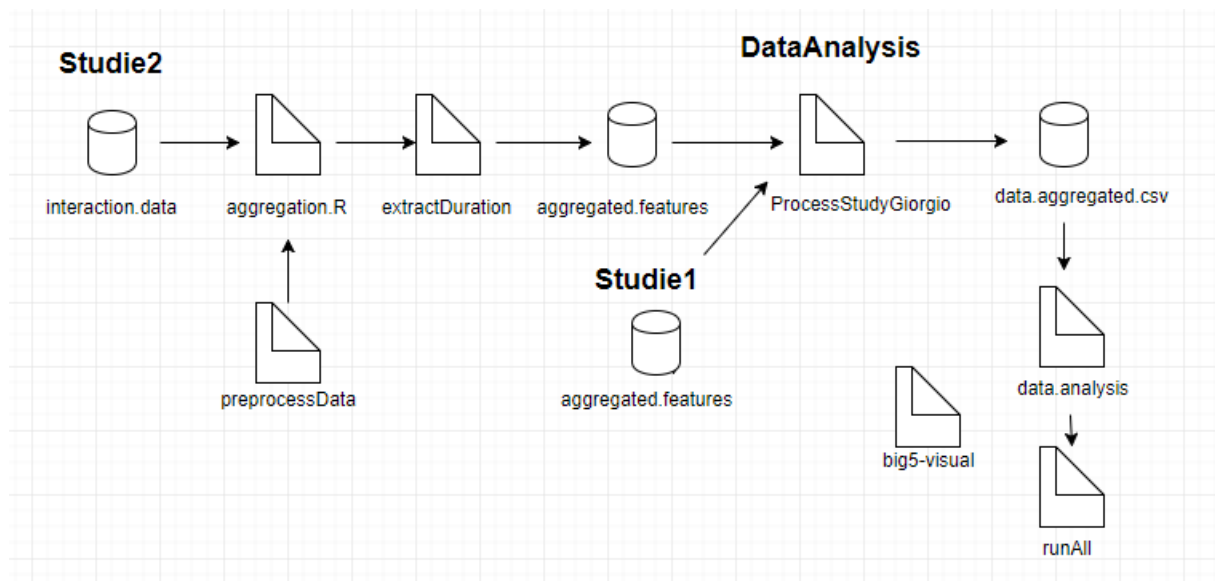


Three folders can be found, with various code-files that are connected like this:



Study1

-- AggregateFeatures.R --

Load aggregated data Study1

Filter out Spell game and 'normal' condition

Prepare set for later merge with Study2:

- create 'Handedness'
 - manually code 'condition' variable following the standard of Study2, described in the "Condition explanation table"
- disabled code-parts contain:
- a routine to spot correlations and output correlation, p-value, and lm-coefficient in one table for all detected variables.

Output: AG.csv

Study2

-- aggregation.R --

The code is taken from Carolas aggregation and adapted to work on this new data set. It aggregates InteractionDataStudie01.Rda into higher level features.

In the outsourced code-file

"preprocessData.R" is deactivated in big parts.

UserID=216 is excluded as it is

"the guy who showed up for the wrong study" (and did not complete it).

Note that this file run only, if there are exactly 12 sessions for every subject. Otherwise the big loop in the end crashes.

line 54-83 provides tools for checking the number of sessions per VP, and if necessary removing them.

output: 1) FeaturesStudie_cleaned.Rda

2) AggregatedFeaturesPerSession_cleaned.Rda

Note: The last loop in the code takes around 30 min to run. However, it works and I simply had lunch during that time instead of fixing it.

But for the protocol, in my opinion the problem is:
For each VP the code runs through all 500 sessions. This is unnecessary as every VP owns and needs only a handful of sessions, and not all.
The code calculates 497 values for every VP, which are not needed and later discarded.

-- extract_duration.R --
for some reason the initial duration-computation does not work on the Study2 data.
this code a) calculated the sessions durations,
b) manually fixes some sessions which were 'split in two': because the App sometimes crashed, Giorgio let the subject play the remaining time in what is counted a new sessions. These few cases are manually merge into one session.

Data Analysis

-- ProcessStudieGiorgio.R --
load the two aggregated features set (studie1 and studie2)
divide the 'count' variables by the session length.
unify columns names, columns and variable-types between the two data sets.
output: data_aggregated.csv --> contain both data sets.

-- DataAnalysis.R --
throw out conditions that are not 'normal' scenario.
dimensionality reduction with 0.8
interpolation of missing values in Big5 with means.
several handy loops for inspecting the data.

-- runAll.R --
the classification.
i found it very cumbersome in R to write functions which take columns of data-frames as arguments. It is therefore much easier to use the "search" toolbar on the upper side of the code-window, type "extra" and in the replace window e.g. "neuro".
Once the code is written for one dimension it can be copy-and-pasted this way within few seconds.
the example works with extraversion.
- code sets up table which neatly reports confidence intervals, significance, etc for all classifiers.
- target class is made.
set.seed operators make sure results are comparable cross classifiers.

-- big5 visual --
number of plots for the dimensions.