# Title

A custom analysis pipeline for brain data using Statistical Parametric Mapping software

# Authors

A. Vorreuther, A. Kasparbauer, U. Ettinger

# Abstract

The following short paper provides information about an analysis pipeline designed to analyze and display results pertaining to behavioral and hemodynamic data. Subjects performed two learning tasks, a procedural learning task, and an object location association task. Code was written using Matlab and IBM SPSS 25 and can be found [here](https://github.com/acv132/UniBonnPraktikum2022).

# Introduction

## Purpose

The given scripts was supposed to serve as an analysis pipeline for behavioral and hemodynamic functional magnetic resonance imaging (fMRI) data of a sample population performing two distinct learning tasks while being administered either of the two substances methylphenidate and nicotine, or a placebo. Using the provided code for analysis was meant to automate the process of obtaining results and make replication possible more easily. Subjects were healthy right-handed male non-smokers (*N* = 75). The first task consisted of a procedural learning (PL) task where subjects were asked to follow a moving asterisk on a screen in a blocked design with random and pattern trials (for details see SOURCE). The second task was an object location association (OLA) task with two phases. During the first phase, images of either artificial or natural objects were shown in one of four quadrants on a screen (encoding phase). The same images were presented together with a set of novel images during the second phase and subjects were asked to recall whether the image was generally familiar and in which quadrant it was presented ([spatial context] retrieval phase; for details see SOURCE).

## Behavioral data

All analyses were written in Matlab (SOURCE) and IBM SPSS (SOURCE). The available code is structured in two pipelines: behavioral data analysis and analysis of fMRI brain data. The first pipeline starts by creating a data frame holding relevant task variables. Descriptive statistics are collected for included subjects (N = 71). Optionally, a chi-square criterion is calculated to exclude subjects from the OLA analysis that statistically only guessed the correct spatial context rather than retrieving it. Sedation ratings assessed by [a questionnaire, SOURCE] are processed and compared between three time points of measurement (pre-medication, pre-scanning, post-scanning).   
 Subsequently, PL and OLA behavioral data is analyzed. For the PL task, mean reaction times (RTs), standard deviations (SDs), and coefficients of variation (CVs) of random and pattern trials were investigated using repeated measures analysis of variance (ANOVA) with trial type and block as within-subject factors (for details see SOURCE). A possible association between the amount of PL and sedation ratings was examined with correlational analysis (Pearson’s r).   
 For the OLA task, encoding and retrieval phase were analyzed separately. For the encoding session, three event types were defined, consisting of two effects of interest, correctly and falsely encoded spatial context, respectively, (CorSCE, FalSCE) and one effect of no interest (comprising items presented during encoding but which were classified as “new” during retrieval as well as missed responses in the encoding and retrieval sessions). Similarly, three event types were defined for the retrieval session: two effects of interest (CorSCR, FalSCR) and one effect of no interest (including items shown during encoding but which were falsely attributed to be “new”, new items correctly or incorrectly responded to, and missed responses). A two-way repeated measures ANOVA was then performed on RTs for effect of substance, subsequent source judgments, or interaction between them. Additionally a signal detection analysis was applied to investigate substance-related differences in “old” versus “new” judgments. This analysis determined the sensitivity *d'* and the response bias *c* and *β* of classifying old items as “old” (for details see SOURCE).

## FMRI data

Further, the available code offers the possibility to run a scripted analysis of fMRI brain data with the Statistical Parametric Mapping (SPM) toolbox (SOURCE). Adjustments to analyses were made possible by configuration options implemented in functions pertaining to each analysis step. First, subjects’ hemodynamic data was preprocessed in a standard procedure including slice-realignment, slice time correction, co-registration and segmentation of images, normalization, and smoothing. After completion of all preprocessing steps, a first-level analysis was performed for both tasks. A full-factorial model was then estimated and an ANOVA was performed for the PL task, OLA encoding, and OLA retrieval phase, respectively (see Behavioral data for details). Six head movement parameters were included as confounds. Further, a pipeline was constructed to display and save results with options to overlay different masks based on contrasts defined during analyses or regions of interest (ROIs). Finally, using the marsbar toolbox (SOURCE), it was investigated what percentage of signal change occurred in specific ROIs during the individual tasks. The code allows for definition of a ROI from a given file and displays average results in form of bar graphs showing the signal change in each substance group.

## Requirements

The pipeline was written on a Windows system and requires IBM SPSS 25 (SOURCE), Matlab SPM12 (SOURCE), and related toolboxes marsbar (SOURCE) and WFU brain atlas (SOURCE)

# Developer Access, Licensing and Availability

# Conclusion

The provided code can be used as an analysis pipeline for the tasks at hand. Both behavioral and brain data was analyzed successfully. Analysis steps can be repeated easily and by systematically modifying provided configuration options in the script, steps can be altered without use of the SPM interface. It should be noted that, while this code is clearly *not* written to provide a general SPM analysis pipeline of fMRI data, individual functions and the overall structure might be useful in developing new pipelines and possibly a more universally applicable script.

# Acknowledgements

# References

# Appendix

Usage examples