The provided scripts can be used to predict the neuroimaging data using behavioural data as inputs. For each given voxel, the neuroimaging data for a specific modality is predicted (using a leave-one-out process) from the full set of behavioural data. This is done by constructing a linear regression model (using support vector regression) at each voxel, where the neuroimaging data is the dependent variable and the behavioural data are independent variables.

The inputs:

1. MAT files that are produced by our pre-processing pipeline (one MAT file per subject);
2. An Excel spreadsheet that contains the behavioural data that will be used as predictors.

The outputs:

1. The NIFTI brain map (which can be opened as an overlay in MriCro software) with voxelwise values of prediction accuracy (Pearson’s correlation coefficient between the actual and predicted values of neuroimaging data, computed across participants);
2. The NIFTI brain map of *p* values corresponding to each voxel’s correlation value from the prediction accuracy map (these values can be used for thresholding);
3. A set of NIFTI brain maps, one for each behavioural predictor, which show the importance of this predictor for predicting the neuroimaging data. They are *Z*-scored beta maps, computed independently for each voxel.

The preparation

1. Locate the folder with the MAT files. Decide on the modality that you would like to analyze; make sure you have its exact name (you can get the list of modalities by running NiiStat). The MAT files that don’t contain the given modality will be ignored in all subsequent analysis.
2. Locate the Excel spreadsheet with behavioural data. The file must contain a sheet with the name “to\_analyze”; the other sheets will be ignored. Make sure that there are no missing values. Only numerical scores that will be used for analysis; anything containing strings will be ignored. Numerical scores that don’t vary across participants will also be ignored.
3. Locate the folders with NiiStat and SPM (we will use some of their functions). Make sure these folders are in the Matlab path; alternatively, add the paths to NiiStat and to SPM to the code of process\_chunk.m and combine\_chunks.m (currently they are commented out).
4. Install this repository on your computer, either by downloading the zipped file (<https://github.com/grigori-yourganov/voxelwise_prediction/archive/refs/heads/main.zip>) and then unpacking it, or by cloning it in the terminal (“git clone https://github.com/grigori-yourganov/voxelwise\_prediction.git”).

The stages of the process:

1. Extracting the relevant modality from subject-specific MAT files (this could be done on a laptop/desktop)
2. Voxelwise prediction of the neuroimaging modality from the behavioural data: this is quite slow and ideally should be done on a cluster. In order to make it more manageable, the brain is separated in eight “chunks” which are analyzed separately. After a chunk is processed, the program produces voxelwise maps of prediction accuracy, the associated *p*-value, and *Z*-maps for each behavioural score.
3. After all chunks are processed, we combine the maps from all chunks into whole-brain maps.

Detailed procedure for each stage

1. Extracting the relevant modality from subject-specific MAT files

This is a preliminary stage, its purpose is to speed up the voxelwise prediction. Here, we use the program loaddata.m to load the subject-specific MAT files, extract the relevant modality, and save it as a single MAT file.

Open the source code for loaddata.m, and specify the path to the MAT files and the relevant modality (e.g., i3mT1). Make sure the name of the modality is exactly as it appears in MAT files (e.g., ‘i3mt1’ won’t work); if in doubt, run NiiStat and write down the exact modality names from the main NiiStat window. Then, run loaddata.m. It will create a MAT file with the same name as the modality, in the same folder as subject-specific MAT files. This file will be used for the next stage.

1. Voxelwise prediction

After the modality is extracted, we can perform voxelwise prediction of the neuroimaging data from the behavioural data using the script process\_chunk.m. The process of voxelwise prediction can be quite slow, and, in order to make it more manageable, the brain is separated in eight “chunks” by cutting through the middle in the top vs bottom, front vs back, and left vs right directions. Prediction is done on each chunk separately. Since prediction is independent across spatial locations, different chunks can be processed in parallel (this is where the cluster becomes really useful). After a chunk is processed, the program produces voxelwise maps of prediction accuracy, the associated voxelwise *p*-value, and *Z*-maps for each behavioural score for the voxels in that particular chunk.

The syntax of calling process\_chunk.m is as follows:

process\_chunk (chunk\_idx, in\_file, out\_rootfolder, behav\_filename)

The arguments are:

chunk\_idx: the number of the brain chunk (0: process whole brain; 1: bottom back right; 2: bottom back left; 3: bottom front right; 4: bottom front left; 5: top back right; 6: top back left; 7: top front right; 8: top front left)

in\_file -- the name of the file containing neuroimaging data;

out\_rootfolder -- the name of the output folder (the process will create a subfolder inside the out\_rootfolder for the specified chunk);

behav\_filename -- the name of the Excel spreadsheet with behavioural data (must contain a sheet called "to\_analyze").

Here is an example:

process\_chunk (1, 'i3mT1.mat', 'i3mT1\_results', 'baseline\_imputed.xlsx')

The process\_chunk.m script calls a function called svr\_core.m (provided in this repository). This function performs leave-one-out SVR prediction.

I have provided an example of how to run process\_chunk.m on a cluster so each chunk gets processed on a separate compute node (see the files main\_batch.sh and qsub\_chunk.sh). This example is written for PBS scheduler; if he cluster uses a different scheduler (e.g. SLURM), the scripts need to be adjusted.

1. Putting results together

If you have called process\_chunk.m with chunk\_index set to zero, it will process the entire brain rather than a chunk. Otherwise, if the brain is processed in chunks, we need to put the chunk-specific results together. This part can be done either on the cluster or on the local computer (it is not computationally intensive).

The script that combines the eight chunks into a whole-brain image is called cobmine\_chunks.m. Before you run it, you will need to set the homepath variable inside combine\_chunks.m so it points to the folder that contains eight subfolders, one per chunk. Then, simply run the combine\_chunks.m script; it creates a new folder inside homepath that contains all the brain maps combined from eight chunks. The folder will contain the following files: r\_map.nii (voxelwise prediction accuracy, computed as Pearson’s correlation between actual and predicted voxel values); p\_map.nii (the corresponding *p* values); and z1\_map.nii, z2\_map.nii, etc. (*Z* values for the first, the second, etc. behavioural scores).