Modeling Toolkit User Manual v1.0

1. Overview

This toolkit contains implementations of mass-univariate and multivariate modeling approaches for inferential and predictive modeling applications. These modeling approaches can be applied to lesion and other high-dimensional imaging datasets.

The modeling approaches that are currently supported in the toolkit currently include partial least squares (regression and classification), regularized (i.e., LASSO, ridge) linear models (regression and classification), support vector machines (regression and classification), ensemble models (regression and classification), mass-univariate Pearson correlation (continuous outcome and/or continuous predictors), mass-univariate t-tests (categorical predictors, continuous outcome), mass-univariate logistic regression (categorical outcome), and mass-univariate Brunner-Munzel tests (categorical predictors, continuous outcome).

While the toolkit is designed for analyses of lesion and lesion-derived data, it can work with any input dataset that is appropriate for regression or classification analyses.

2. Setting up the Toolkit

Download and unzip the repository.

Once you have unzipped the repo, you should open MATLAB R2022b (or a later version) and navigate to the /mlsm_code_repo folder containing the unzipped repository.

3. Tutorial Notebooks

Within the mlsm_code_repo folder, there is a folder labeled “tutorials”. Within this folder, there are subfolders “multivariate_regression”, “multivariate_classification”, “mass_univariate”, and “model_stacking”. Within each of these subfolders, there are tutorial notebooks for different modeling methods (e.g., partial least squares, support vector machines, etc.) and predictor data modalities (e.g., lesion data contained in NIFTI images, disconnection data contained in adjacency matrices, etc.). These tutorial notebooks provide step-by-step instructions for running different types of analyses with the toolkit.

4. Running Lesion Modeling Analyses with the GUI

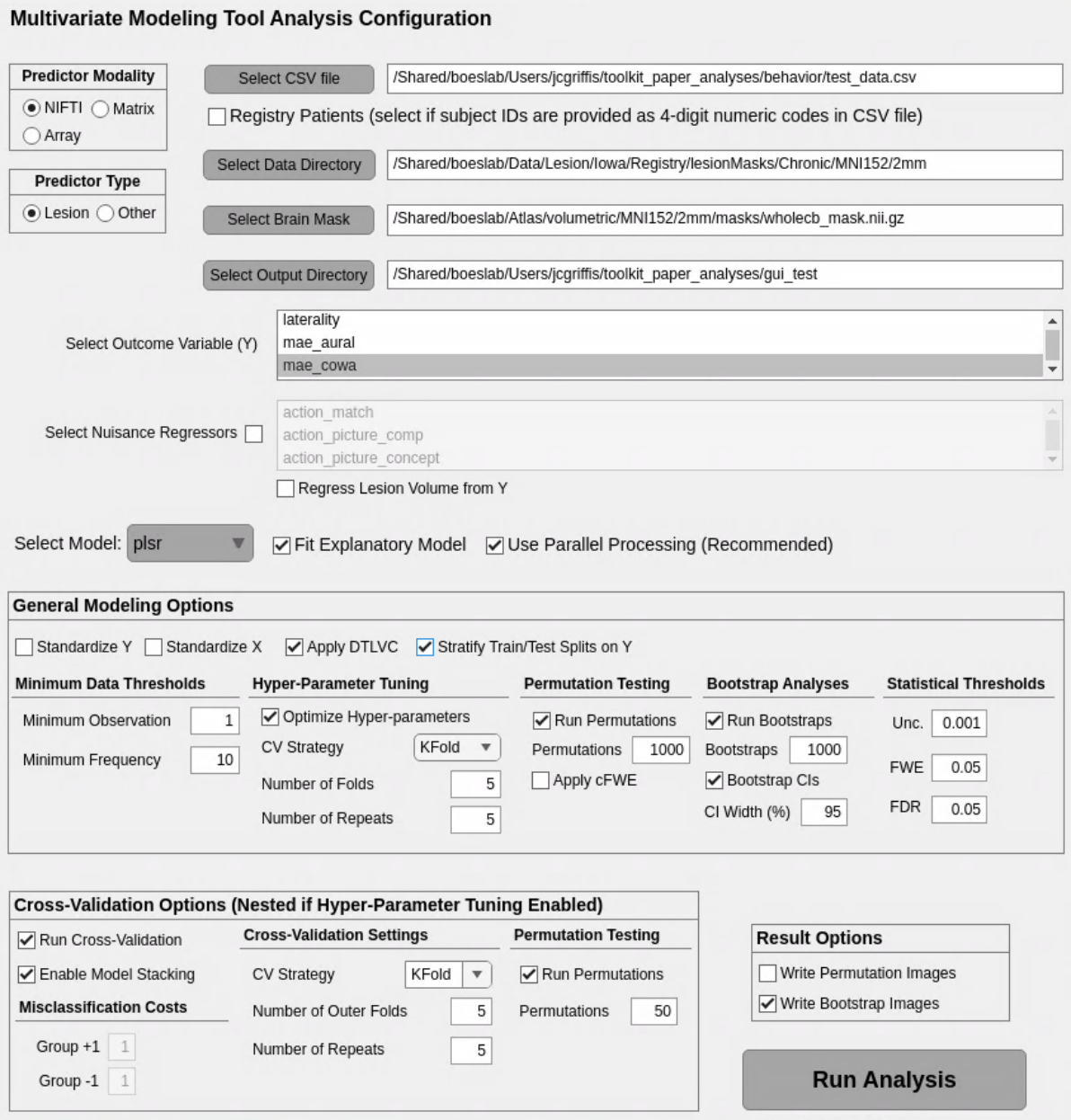
In addition the scripting interface described in the above sections, the toolkit also features a graphical user interface (GUI).

To start the GUI, navigate to the unzipped copy of the repository in your User directory, and then type:

run_modeling_gui.m

into the MATLAB Command Window.

This will activate the GUI in a new MATLAB app window:



The GUI provides a simplified interface for configuring and running analyses.

1. Select the appropriate button in the **Predictor Modality box** located in the upper left hand corner. If your predictor data are NIFTI images (e.g., lesion masks or lesion-network maps stored as .nii or .nii.gz files), then you should select the “NIFTI” option. If your predictor data are connectivity matrices stored in .txt or .csv files, then you should select the “Matrix” option. If your predictor data are array data (e.g. parcel-level measures) stored in .txt or .csv files (i.e., one variable per column), then you should select the “Array” option. Currently, these are the only types of predictor inputs that are supported by the GUI. Any arbitrary predictor inputs can be accommodated using the scripting interface so long as they (1) are provided as an N_subject-by-N_predictor matrix, and (2) the number of rows in the predictor matrix is equal to the number of observations in the outcome variable.

2. Select the appropriate button in the **Predictor Type box**, located directly under the Predictor Modality box. If your predictor data are lesion masks, then you should select the "Lesion" option. If they are any other type of data (e.g., lesion-network maps), then you should select the "Other" option.
3. Click on the **"Select CSV file" button** to the right of the boxes described above. This will open a dialogue window that you can use to select a CSV file formatted so that:
    a. The first column is named "study_id" and contains patient identifiers that correspond to file pre-fixes for your predictor data (e.g., if your predictors are lesion images named like "0018.nii.gz", then the "study_id" field for this patient would be 0018).
    b. The outcome measure of interest is contained in another column. Select that column from the column names shown in the box labeled "Select Outcome Variable (Y)".
    c. Nuisance regressors of no interest are contained in other columns (if relevant). If you have nuisance regressors in other columns that you would like to have regressed from the outcome data prior to fitting your model, then select the "Select Nuisance Regressors" checkbox, and select the field(s) corresponding to the nuisance regressor(s). If the predictor data are binary voxel-based lesion data, then you may also select the "Regress Lesion Volume from Y" checkbox; if checked, the software will automatically estimate the lesion volume for each patient from their lesion image and regress the lesion volume estimaes from the outcome variable prior to fitting the model. Note – any categorical covariates must be dummy-coded as binary (1 or 0) variables.
4. If your CSV file's "study_id" field contains numeric versions of 4-digit Registry IDs (e.g., for patient "0018", the data type is numeric such that it evaluates to "18"), then click the **Registry Patients button** located below the "Select CSV file" button. This will allow the toolkit to appropriately handle these IDs when constructing filenames to load your predictor data. If the "study_id" field contains any other type of ID (e.g., clinical IDs), then these IDs need to exactly match the filename prefixes of your predictor data (e.g., for patient "ca001", the corresponding lesion filename should be "ca001.nii.gz"), and you should not select the **Registry Patients button.**
5. Click the **Select Image Directory button** located beneath the Select CSV File button. This will open a dialogue window that will allow you to select the directory containing your predictor images (e.g., the directory containing lesion masks).
6. Click the **Select Brain Mask** button located beneath the Select Image Directory button. This will open a dialogue window that will allow you to select the directory containing a **brain mask** that is registered to the same space and has the same image and voxel dimensions as your image predictors.
7. Click on the **Select Model** dropdown menu and choose the type of model that you want to use for your analyses.
8. If you want to perform inferential lesion-symptom mapping analyses based on a model fit to the full dataset, then click the **Fit Explanatory Model** button next to the Select Model dropdown menu. This will perform a traditional inferential analysis using the full dataset that will return e.g., statistical maps that can be thresholded based on statistical significance. Most of the time, you will want to run this type of analysis.
9. If you are using MATLAB R2022b or later with the Parallel Processing Toolbox, and have sufficient resources for parallel processing, then click the **Use Parallel Processing button** next to the Fit

Explanatory Model button. This will turn on MATLAB parallel processing, greatly reducing the amount of time needed to run the analysis.

10. Select additional modeling options from within the **General Modeling Options box**.
    a. Standardize Y – standardizes the outcome variable
    b. Standardize X – standardizes the predictor matrix
    c. Apply DTLVC – apply the direct total lesion volume control method described by Zhang et al., (2014 – Human Brain Mapping). This is only used for lesion analyses.
    d. Regress Lesion Volume from Y – regresses lesion volume from the outcome variable and performs analyses using the residuals as the outcome variable. This is only used for lesion analyses.
    e. Stratify Train/Test Splits on Y – stratifies cross-validation partitions based on quartiles of the outcome variable.
11. Set **Minimum Data Threshold** options
    a. Minimum Observation - the minimum non-zero observation that is meaningful. For lesion analyses with binary data, this is 1.
    b. Minimum Frequency - the minimum number of data points with values > Minimum Observation required for inclusion of a predictor in the final analyses. For example, if doing a lesion-symptom mapping analysis with binary lesion data, then if you have Minimum Observation set to 1 and Minimum Frequency set to 4, only voxels that are lesioned in at least 4 patients will be included in the analysis.
12. Set **Hyper-parameter Tuning** options
    a. Optimize Hyperparameters – check this box to enable hyper-parameter optimization. This uses cross-validation to determine optimal hyper-parameters for your selected model. If not selected, default values will be used.
    b. CV Strategy – this determines the type of cross-validation used for hyper-parameter optimization. Options are KFold and LOOCV (leave-one-out).
        i. Number of Folds – if KFold, then this is the number of folds
        ii. Number of Repeats – if KFold, then this is the number of repetitions
13. Set **Permutation Testing** options
    a. Run Permutations – check this box to run permutation tests for statistical significance. Recommendation for most analyses is to run permutations (1000 or more) to test for significance of model fit, and then use bootstrapping (see next section) to estimate significance of model coefficients.
        i. Permutations - the number of permutation iterations to run. Minimum possible p-value obtainable is given by: 1 / (number of permutations + 1). I.e., for 1000 permutations, the minimum possible p-value is 0.0009.
        ii. Apply cFWE – use the continuous family-wise error method for thresholding statistical maps (Mirman et al., 2018 – Neuropsychologia). Recommended for mass-univariate models.
14. Set **Bootstrap Analyses** options
    a. Run Bootstraps – check this box to run bootstrap analyses to generate confidence intervals and/or test statistics on model coefficients. Only applies to multivariate models.
        i. Bootstraps – the number of bootstrap iterations to run. 1000 is recommended as a minimum, but 5000+ is ideal.

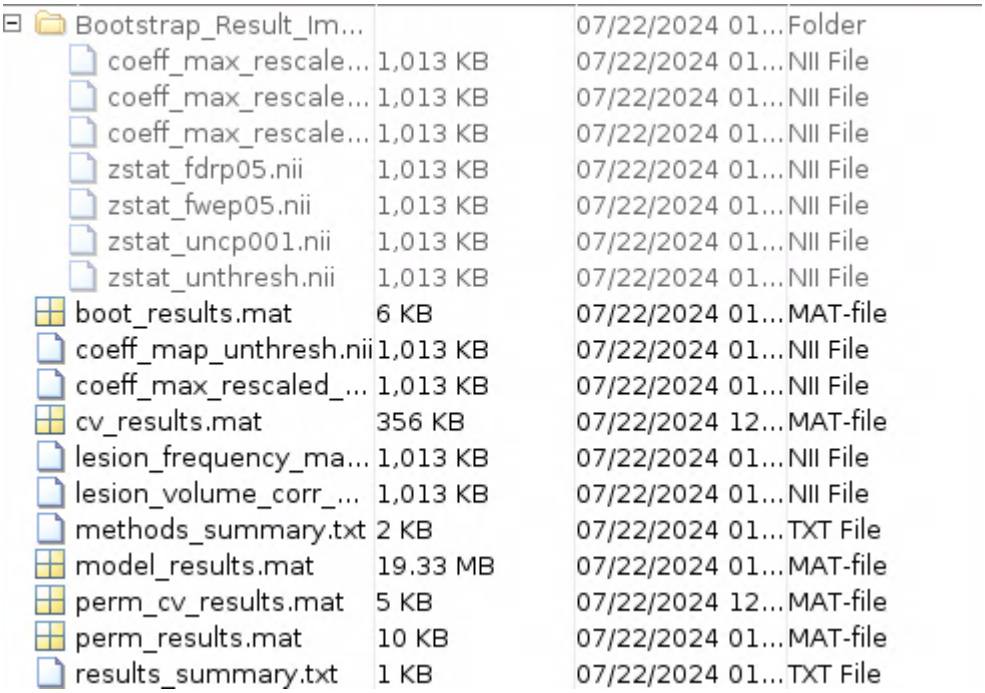
ii. Bootstrap CIs – check this box to compute confidence intervals on the model fit and coefficients based on the bootstrap estimates.
    1. CI Width – the coverage of the CI (e.g., 95 corresponds to a 95% confidence interval). Note: CIs are not corrected for multiple comparisons.

15. Set **Statistical Threshold** options
    a. Unc. - this corresponds to the uncorrected p-value threshold for writing out result images.
    b. FDR – this corresponds to the False Discovery Rate corrected p-value threshold for writing out result images.
    c. FWE – this corresponds to the Family-Wise Error corrected p-value threshold for writing out result images.
16. Set **Cross-Validation** options
    a. Run Cross-Validation – check this box to run cross-validation analyses to evaluate the out-of-sample predictive power of your model. Only select this option if your goal is to run a predictive analysis.
    b. Enable Model Stacking – check this box if you wish to stack cross-validated models of the same behavioral outcome. For example, if you run lesion, structural disconnection, and functional lesion-network models of the same behavioral outcome, then you may want to evaluate whether the outputs of these models can be combined to improve predictive power. This can be achieved by model stacking. If you choose to enable model stacking, then no patients will be dropped from the model, and the same cross-validation partitions will be used across all models of the same outcome to allow for subsequent stacking. Model stacking can be performed using a separate GUI that can be invoked by typing run_stacking_gui into the MATLAB command window from within the modeling toolkit directory (see **section 6**).
    c. Misclassification Costs – set misclassification costs for each group. By default, misclassification costs for both groups are equal, but this can be modified to adjust for e.g., highly imbalanced classes.
    d. Other options correspond to same options as in earlier portions of GUI, but control analogous options within the cross-validation framework.
17. Set **Result Options**
    a. Write permutation images – check this box to write out result images thresholded based on permutation testing. Only check if permutation testing is enabled and Apply cFWE box is also checked.
    b. Write bootstrap images – check this box to write out result images thresholded based on bootstrap analyses. Only check if bootstrap analyses are enabled.
18. Click on the **Select Output Directory** button at the bottom left of the GUI. This will open a dialogue box that you can use to select the directory where you want to save the results.
19. Click the **Run Analysis** button in the bottom right-hand corner of the GUI. This will run the analysis based on the options you have selected.
20. See **section 5** for information about the imaging result outputs, and see the MATLAB live script **lesion_symptom_modeling_tutorial.mlx** in the main toolkit folder for information about the

.mat files **model_results.mat** and **cv_results.mat** that contain the outputs of inferential (explanatory) and predictive (cross-validated) models, respectively.

5. Imaging Outputs for Explanatory Models

Depending on the specific parameters and options chosen for your analysis, the output directory for a typical inferential analysis using explanatory models will look something like this for multivariate models:



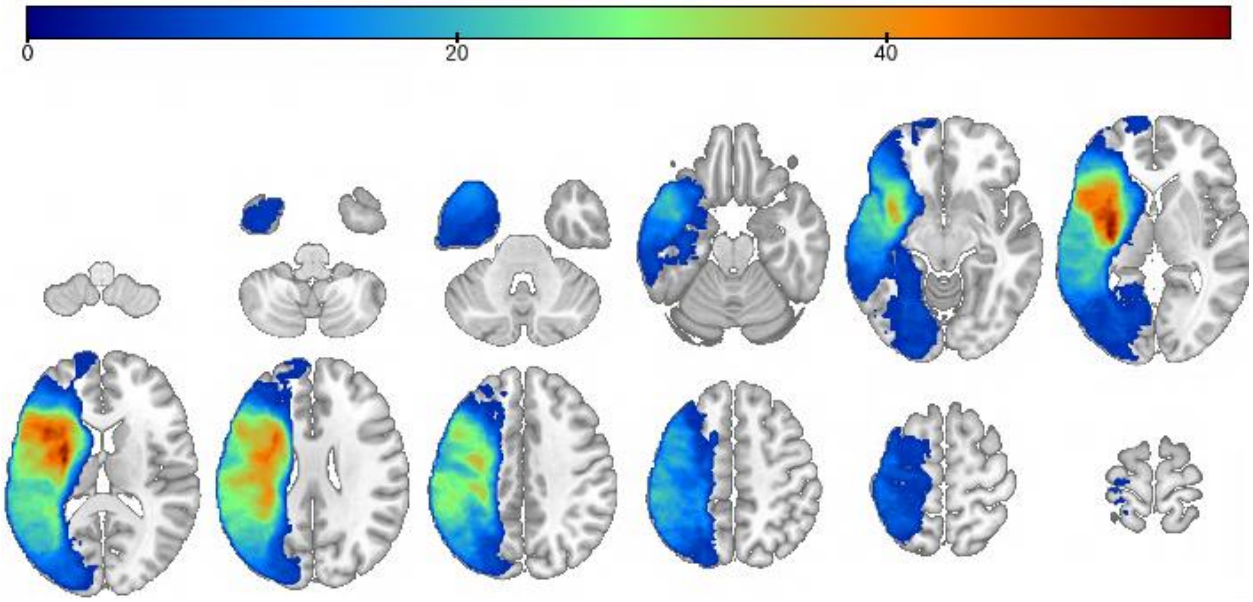
or like this for mass-univariate models:

Permutation_Result_...		07/17/2024 10...	Folder
coeff_fwe05_vcrit_...	1,013 KB	07/17/2024 10...	NII File
coeff_fwe05_vcrit_...	1,013 KB	07/17/2024 10...	NII File
coeff_fwe05_vcrit_...	1,013 KB	07/17/2024 10...	NII File
coeff_fwe05_vcrit_...	1,013 KB	07/17/2024 10...	NII File
coeff_fwe05_vcrit_...	1,013 KB	07/17/2024 10...	NII File
coeff_fwe05_vcrit_...	1,013 KB	07/17/2024 10...	NII File
coeff_vfdr05.nii	1,013 KB	07/17/2024 10...	NII File
coeff_vfwe05.nii	1,013 KB	07/17/2024 10...	NII File
tstat_vfdr05.nii	1,013 KB	07/17/2024 10...	NII File
tstat_vfwe05.nii	1,013 KB	07/17/2024 10...	NII File
coeff_vfdr05.nii	1,013 KB	07/17/2024 10...	NII File
coeff_vfwe05.nii	1,013 KB	07/17/2024 10...	NII File
corr_map_unthresh.nii	1,013 KB	07/17/2024 10...	NII File
lesion_frequency_m...	1,013 KB	07/17/2024 10...	NII File
lesion_volume_corr_...	1,013 KB	07/17/2024 10...	NII File
methods_summary....	2 KB	07/17/2024 10...	TXT File
model_results.mat	1.07 MB	07/17/2024 10...	MAT-file
results_summary.txt	1 KB	07/17/2024 10...	TXT File
tstat_map_unthresh...	1,013 KB	07/17/2024 10...	NII File
tstat_vfdr05.nii	1,013 KB	07/17/2024 10...	NII File
tstat_vfwe05.nii	1,013 KB	07/17/2024 10...	NII File

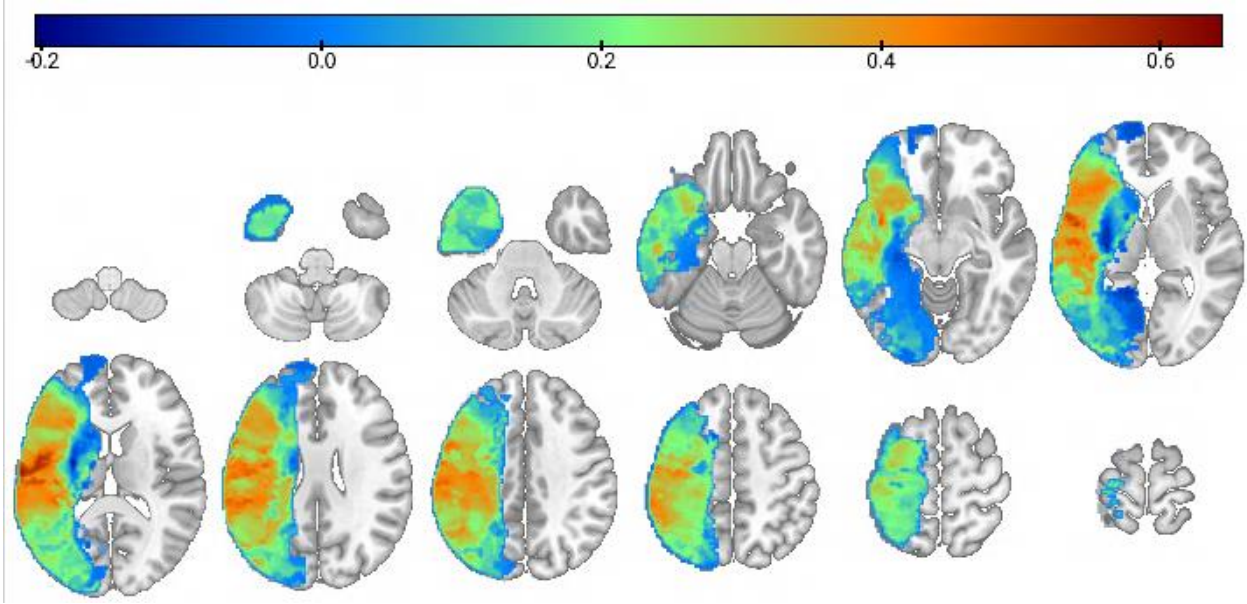
5a. Main Output Folder Images

This folder will contain, for multivariate lesion-symptom mapping analyses, the following files by default:

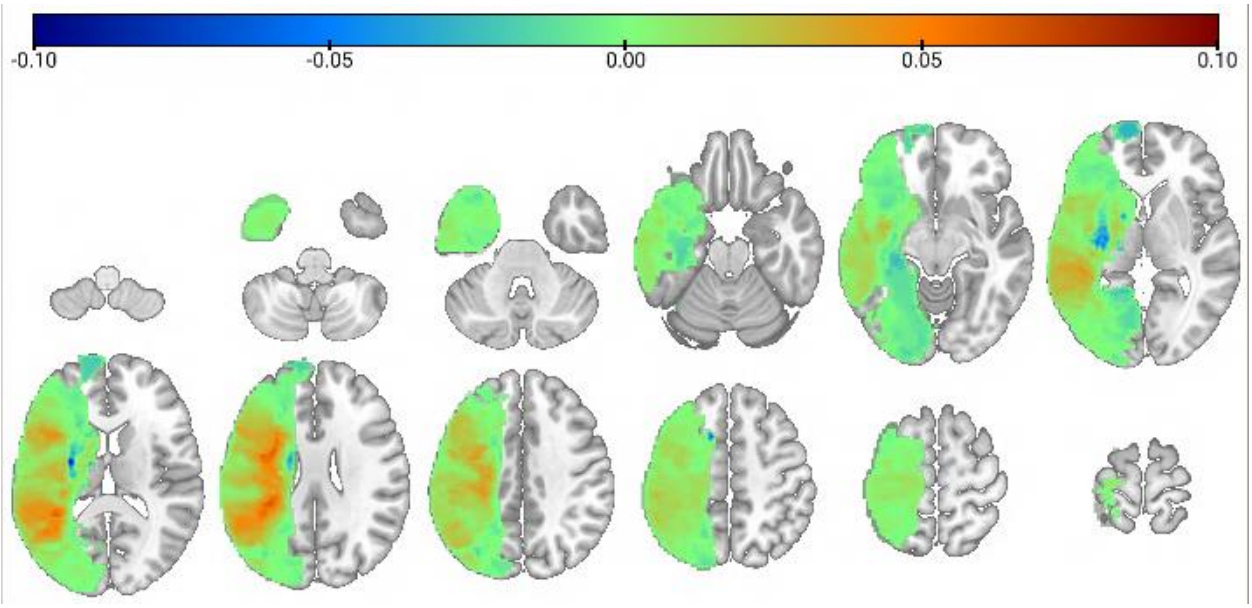
- lesion_frequency_map.nii – this is a frequency map generated by summing all lesion images in the sample. Voxel values correspond to counts indicating how many lesions affect each voxel. An example map is shown below:



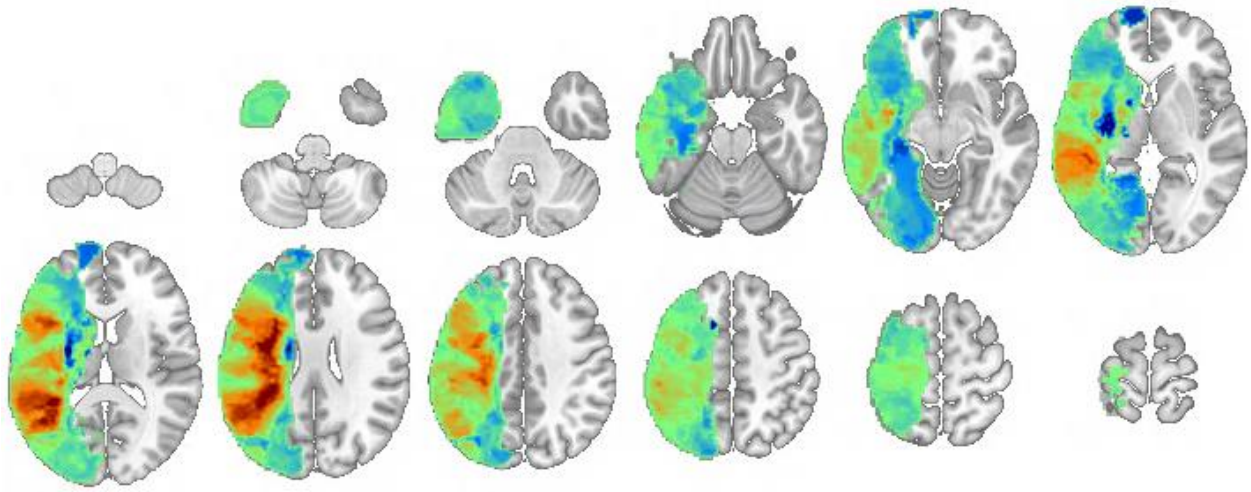
- lesion_volume_corr_map.nii – this is a correlation map generated by correlating voxel lesion statuses with lesion volume. Voxel values correspond to correlation coefficients indicating the strength of the linear relationship between lesion volume and voxel lesion status i.e., larger values indicate voxels that are more commonly damaged in patients with large lesions. An example map is shown below:



- coeff_map_unthresh.nii – this is a model coefficient map obtained from fitting the model specified in the cfg structure. It is not statistically thresholded, and values are in the scale of the original weights, which are often very small (e.g., 0.00001). It can still be useful for understanding the raw model coefficients. An example is shown below:



- coeff_max_rescaled_unthresh.nii – this is the same map as coeff_map_unthresh.nii, but voxel values have been re-scaled as a proportion of the maximum value in the map. An example is shown below:

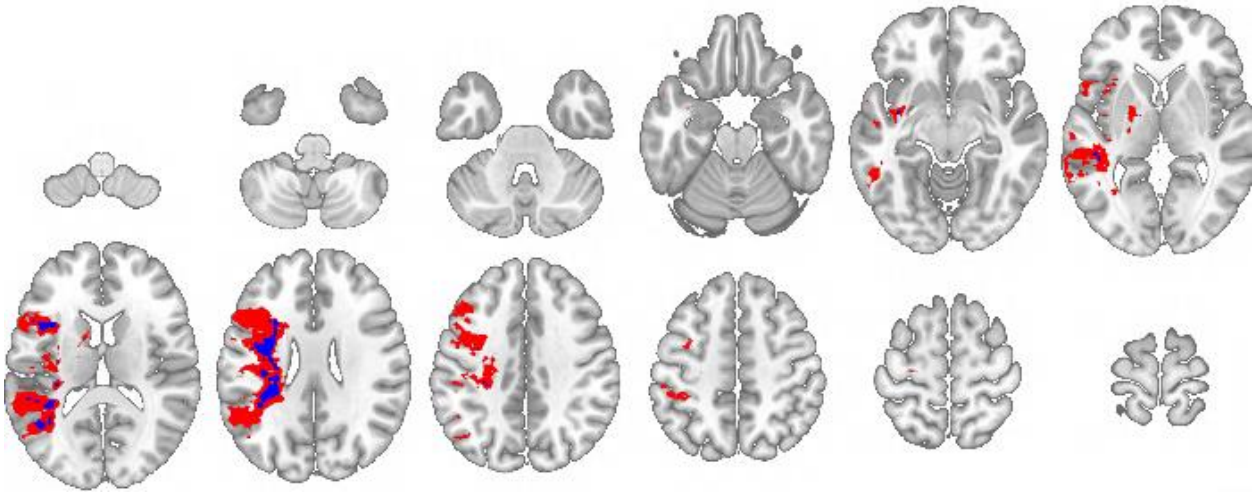


- model_results.mat – this contains the main results for inferential models fit to the full dataset and also contains the "cfg" struct containing analysis parameters and options. See the tutorial notebooks for detailed descriptions of the different fields contained within the model_results struct.
- cv_results.mat - this contains the results of the cross-validation analyses. See the tutorial notebooks for detailed descriptions of the different fields contained within the cv_results struct.

5b. Permutation_Result_Images Subfolder

This folder will contain the results of permutation testing if it was performed with default values in the cfg structure.

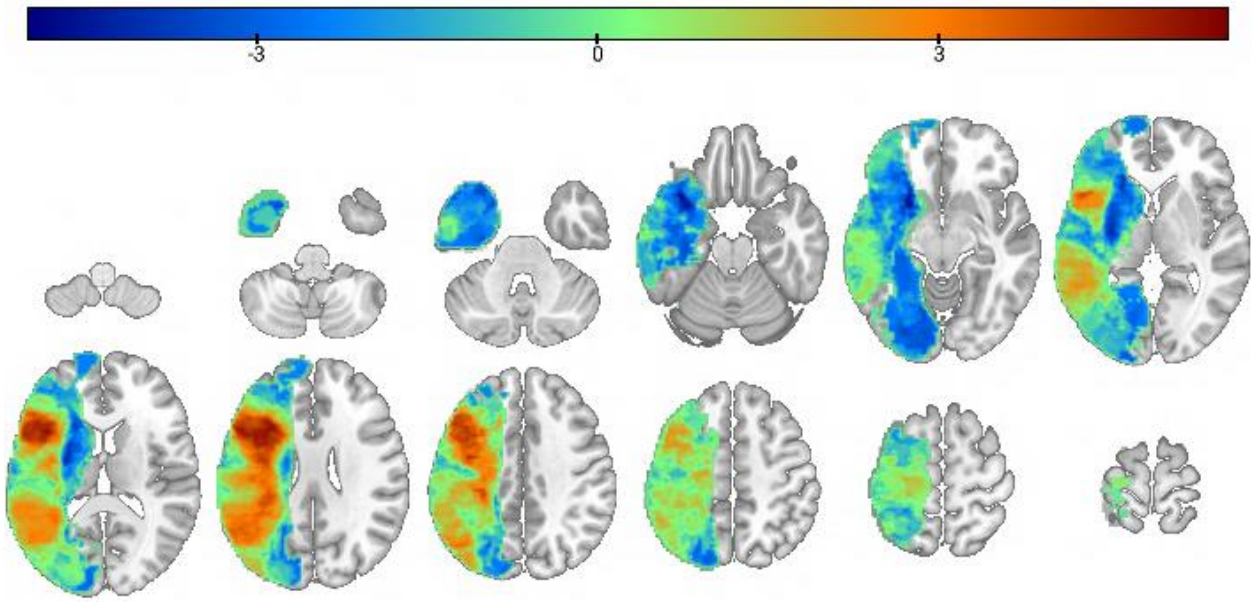
- coeff_max_rescaled_crit_val_x.nii – there is a set of images named like coeff_max_rescaled_crit_val_x.nii where "x" takes values of [1,10,50,100,500,1000]. These images correspond the coeff_max_rescaled_unthresh.nii image described in the previous section, but with continuous family-wise error thresholding applied at different voxel thresholds (Mirman et al., 2017 – Neuropsychologia).



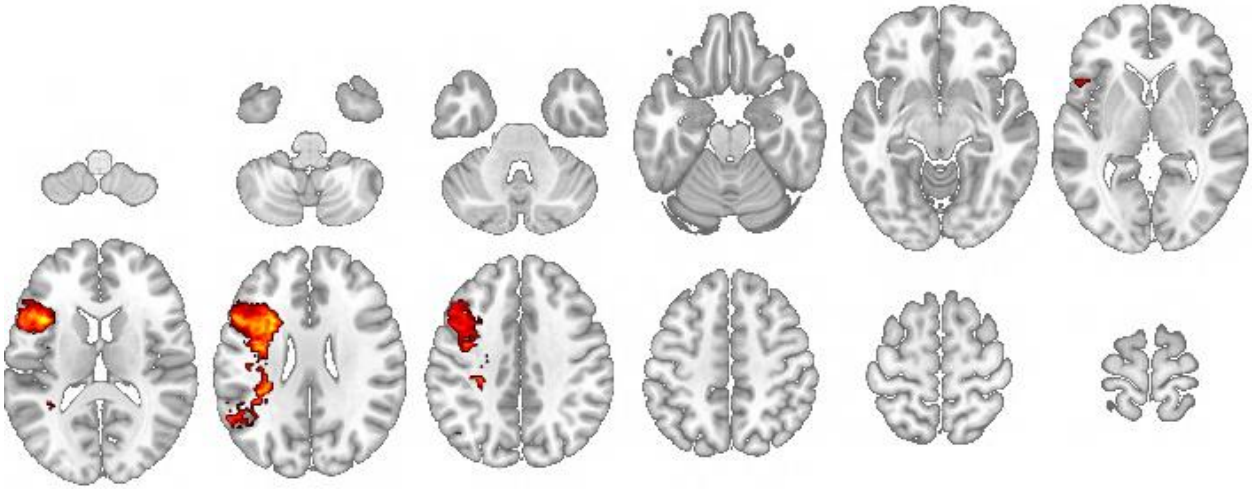
5c. Bootstrap_Result_Images Subfolder

This folder will contain the results of the bootstrap analysis if it was performed with default values in the cfg structure.

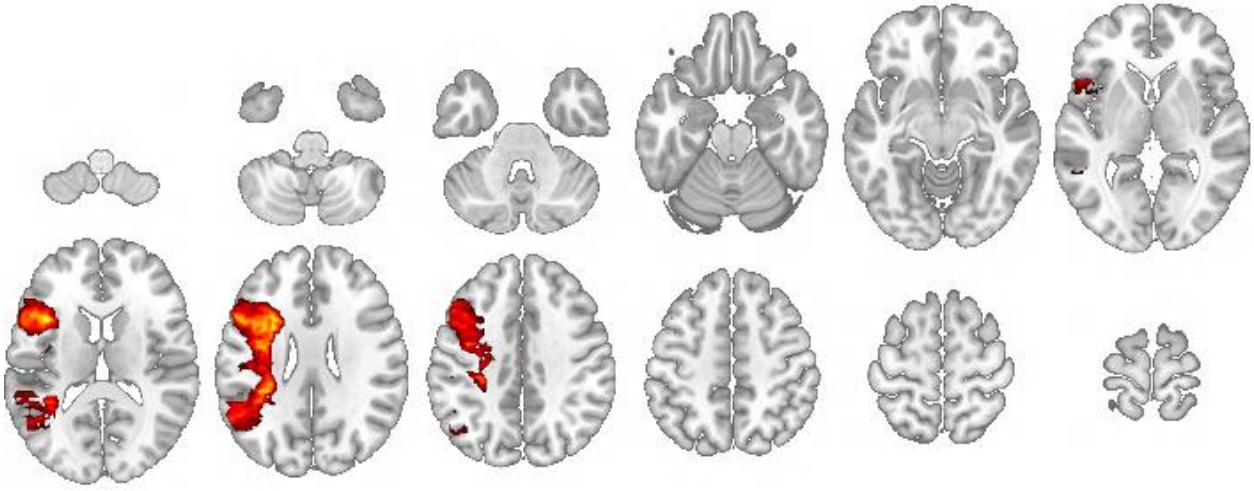
- zstat_unthresh.nii.gz – this image contains unthresholded *z*-statistics for the model coefficients obtained from the bootstrap analysis. These *z*-statistics are test statistics on the model coefficients derived from the bootstrap coefficient distributions according to normal probability theory and are therefore parametric test statistics unlike the non-parameteric permutation-based statistics (i.e., a *z*-statistic of 1.96 corresponds to a 2-tailed p-value of 0.05). This map reflects the stability of the coefficient estimates and the corresponding uncorrected p-values, not the effect sizes (i.e., these are in the coeff_map images in the main output folder), similar to a *t*-statistic map. The *z*-statistic image for the unthresholded coefficient map in the previous section is shown below:



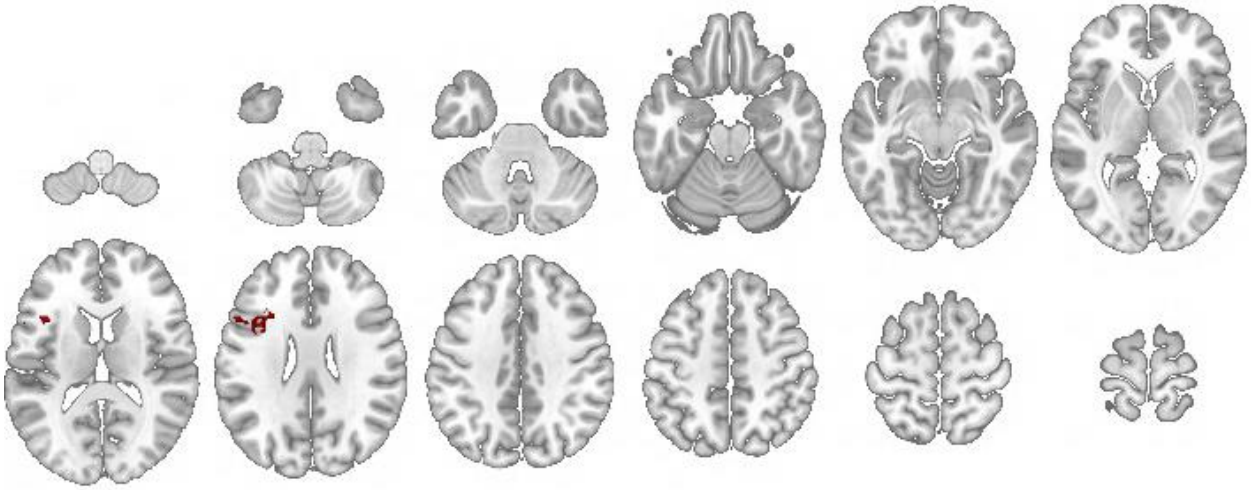
- coeff_max_rescaled_uncp001.nii – if the analysis was run with default settings in the cfg for the bootstrap analysis, then this image will be output. This image corresponds to the coeff_max_rescaled_unthresh.nii image described earlier, but thresholded to only retain voxels with uncorrected 2-tailed p-values < 0.001. An example is shown below:



- coeff_z_rescaled_fdrp05.nii – the analysis was run with default settings in the cfg for the bootstrap analysis, then this image will be output. This image corresponds to the coeff_max_rescaled_unthresh.nii image described earlier, but thresholded to only retain voxels that survive a False Discovery Rate (FDR) correction at 0.05. This is usually stricter than the uncorrected p < 0.001 threshold, but this is not always the case since the FDR correction threshold depends on the observed p-value distribution. An example is shown below:



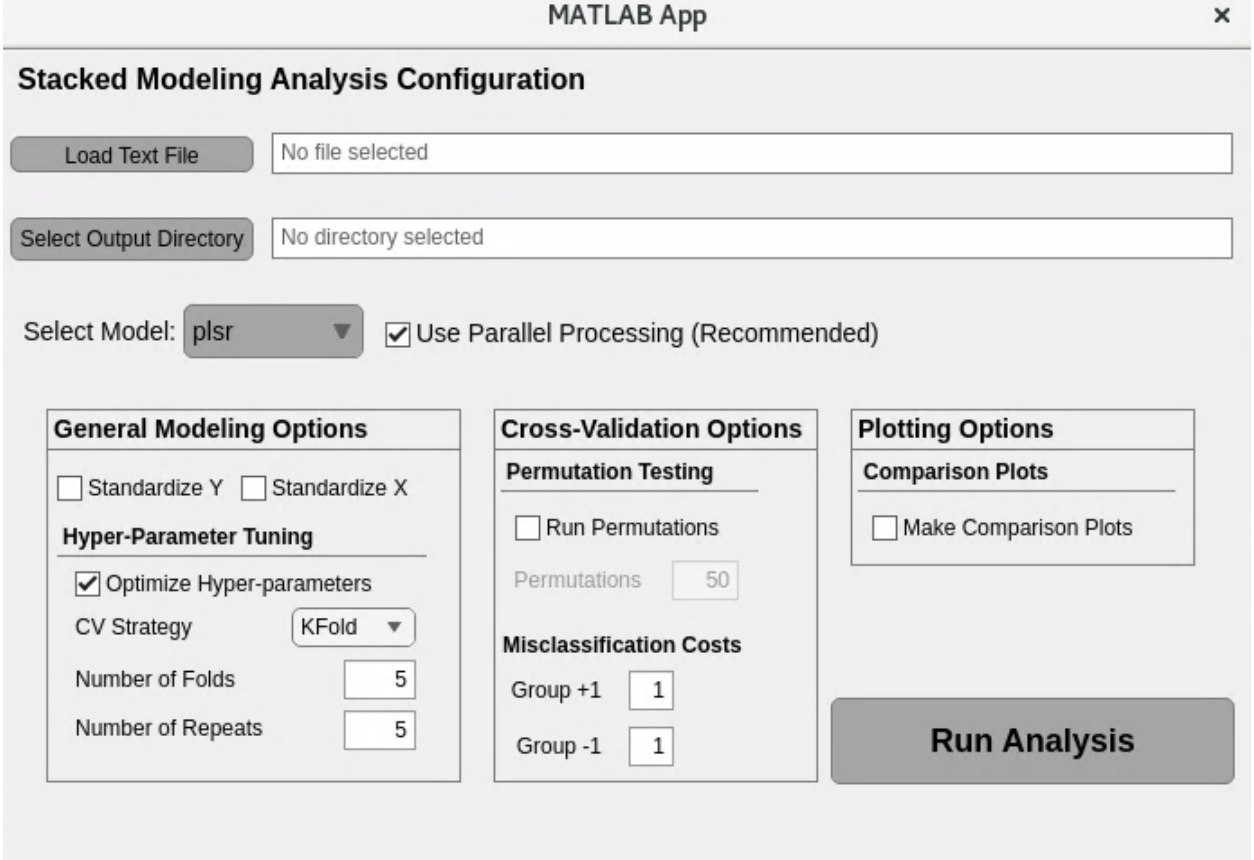
- coeff_z_rescaled_fwep05.nii – the analysis was run with default settings in the cfg for the bootstrap analysis, then this image will be output. This image corresponds to the coeff_max_rescaled_unthresh.nii image described earlier, but thresholded to only retain voxels that survive a Family-Wise Error rate (FWEr) correction at 0.05. This is the strictest threshold. An example is shown below:



6. Model Stacking GUI

Model stacking is an approach to predictive modeling that aims to leverage the unique predictive power of multiple individual "base" models by creating a "meta-model" that aims to predict the outcome of interest by training a new model that uses the "stacked" (i.e. into a predictor matrix) predictions from the individual base models as predictors.

To invoke the model stacking GUI, type run_stacking_gui into the MATLAB command prompt from within the model toolkit directory. This will open a window that looks like the one shown below:

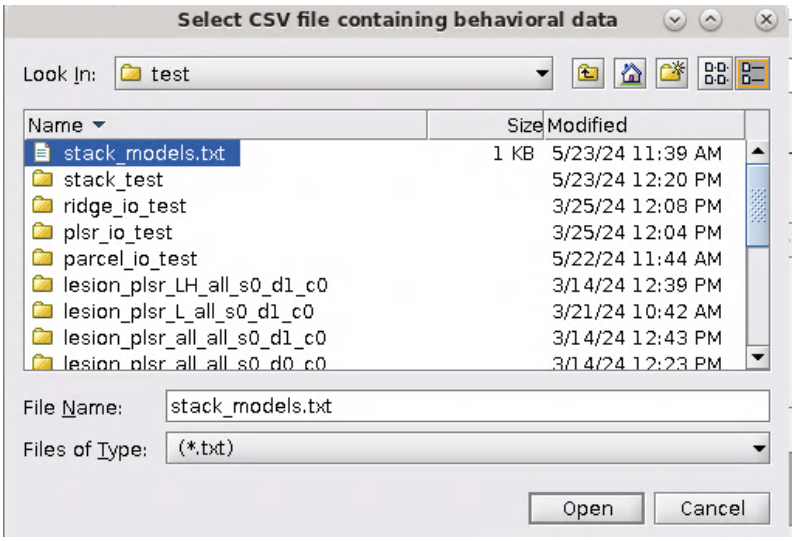


To run stacked models using the GUI, first, create a text file that contains, on each line, a path to the results for a model that you want to include in the final stacked model. For example, the text file shown below includes paths to the results for four different PLSR models of the mae_token outcome that each used different predictor modalities:

```
/Shared/boeslab/Users/jcgriffis/language_project/model_results/plsr/lesion_all_all_s0_d1_c0/mae_token
/Shared/boeslab/Users/jcgriffis/language_project/model_results/plsr/chaco_all_all_s0_d0_c0/mae_token
/Shared/boeslab/Users/jcgriffis/language_project/model_results/plsr/slnm_tdi_all_all_s0_d0_c0/mae_token
/Shared/boeslab/Users/jcgriffis/language_project/model_results/plsr/flnm_all_all_d0_d0_c0/mae_token
```

These models must all use the exact same patient sample and model the same behavioral outcome, and they must be cross-validated using the same cross-validation partitions. This is ensured by selecting the ”Enable Model Stacking” option in the main GUI. Each model must have both a ”cv_results.mat” file containing the cross-validation results, along with a ”model_results.mat” file containing other general model results and parameters.

Click the “Load Text File” button, and select the text file containing the paths to the model results folders for the models that you want to include in your stacked model:



Next, click on the "Select Output Directory" button, and select the folder where you want to output the results of your model stacking analysis.

Finally, select the model that you wish to use for model stacking, set any general modeling options related to variable standardization and/or hyper-parameter tuning, and set any options for permutation testing the cross-validated stacked model as relevant for your intended analysis.

Note, the main cross-validation settings for the model stacking analysis will be identical to those used for the base models.

See the tutorial script **model_stacking_tutorial.mlx** for a full description of the output of the model stacking analyses.

7. Other

Note: I recommend using down-sampled lesion data unless you really think that your results require 1mm resolution. I typically run analyses with 2mm or 3mm isotropic lesion data. Lower-resolution lesion data greatly decreases the computational load, memory requirements, etc. and greatly speeds up processing. It also reduces the severity of multiple comparisons corrections by a large factor since the required thresholds will depend on the total number of predictors. So far, I've seen no advantage to using 1mm lesion data.

Note: By default, parallel processing is turned on. This greatly speeds up performance, but requires more computational resources than running without parallel processing. Parallel processing requires MATLAB 2022b or later (only tested on 2022b so far, though) as it is unreliable on earlier versions. Scripts can be converted to functions or run as stand-alone scripts on Argon, and I recommend running final analyses on Argon especially if you have a large patient sample, want to do nested cross-validation, want to run more than 1000 bootstrap/permutation iterations, and/or are using 1mm data since running parallel processing for large analyses on ITF machines can consume most or all of the resources on the machine. It is very simple to run on Argon, and I have some scripts for generating q-sub jobs via MATLAB that could be modified if you're interested, so reach out if this is something you would like help with.