### **1. Project's Title**

A robust harmonisation approach for cognitive data from multiple ageing and dementia cohorts (cognitive\_harmonisation).

### **2. Project Description**

Although many cognitive measures have been developed to assess cognitive decline due to Alzheimer’s disease there is little consensus on optimal measures, leading to varied assessments across research cohorts and clinical trials making it difficult to pool cognitive measures across studies.

We provide a two-stage approach to harmonise cognitive data across cohorts and derive a cross cohort score of cognitive impairment due to Alzheimer’s Disease. First, we provide code to harmonise cognitive data from international cohorts of varying size and ethnic diversity via k-nn imputation. Next, we provide code to derive cognitive composites that leverage maximal data from the harmonised dataset.

### **3. Required Software and toolbox**

*Software*

MATLAB

v 2019a or greater has been tested, prior releases are likely to work fine but have not been extensively tested.

*Toolboxes (non critical)*

Parallel Computing Toolbox

The impute.m code opens a parallel pool on the local machine with num cores = number of cores available-2. This is because the code calls parfor loops in the search of optimal k neighbours. Without Parallel Computing Toolbox (Distrib\_Computing\_Toolbox) the code will run fine, except it will be considerably slower, particularly in the within cohort imputation if there is a high degree of randomness in missing variables.

**4. How to Install and Run the Project**

1. Download the cognitive\_harmonisation directory.
2. Scale and merge your raw cognitive data (TARGET data), ensuring data are within range and are in the correct format. The scaled and merged data needs to be saved in the data directory.
   1. Prior to imputation data has to be in a specific format and in some instances needs slight augmentation. With this package we provide several examples of the data transformation and augmentation (see section 7. Example Data). This toolbox includes dummy data and exemplar scalemerege\_example.m scripts which provide the basis for new data to be scaled and merged. This is the most hands-on requirement for our approach and as such we provide a standalone description on how users might do this see section 7. Example Data.
3. Run impute.m
   1. User is prompted to enter the name of the scaled data.

>> impute

what is the name of the scaled merged data, e.g. example\_data\_SCALE\_MERGE?

* 1. Enter the file name (i.e., example\_data\_SCALE\_MERGE) and press enter.

>> impute

what is the name of the scaled merged data, e.g. example\_data\_SCALE\_MERGE? example\_data\_SCALE\_MERGE

* 1. User is prompted to enter the hold out ground truth variable. This variable is used to optimise k and also provides a measure of the fidelity of the imputation.

what is the ground truth hold out variable, e.g. ravlttot?

* 1. Enter the ground truth variable (i.e.ravlttot) in the command window. Variable names are found in the variable table (Supplementary Table 1.).

run fast impute?, y/n?

* 1. User is prompted to enter either y or n to indicate whether they would like to run the full harmonisation pipeline (within sample, optimisation of k, imputation from REFERENCE harmonised data into TARGET cohort)
     1. If run fast impute is no (n), this will run through the complete harmonisation pipeline as discussed in the accompanying manuscript. Depending on the access to sufficient RAM/CPUs this may take some time to run through.
     2. If run fast impute is yes (y), this will only run the sequential harmonisation code to impute data from the REFERENCE harmonised data into TARGET cohort. This skips the within sample imputation and sets k neighbours = 3; this value is a fairly safe bet to maintain dispersion of the data and reduce imputation errors. If you have a small sample (n<100), a sample with poor variability (i.e., many of one diagnostic category and not many of another), wish to impute a single visit, or are in a rush, this is the best option.

1. On completion of running impute.m
   1. a full set of harmonised variables will be saved in the data directory as TARGET\_Imputed.mat file and TARGET\_Imputed.csv.
   2. A Figure showing the optimal k value based on RMSE of the hidden ground truth variable for the within cohort imputation is shown. (*if run fast impute is n*)
   3. A Figure showing the optimal k value based on RMSE of the hidden ground truth variable when hiding this score from the full TARGET cohort and imputing it in from the REFERENCE cohort is shown. The RMSE at the selected k value gives an indication of the fidelity of the imputation. RMSE can be seen by pressing the red circle on the plot in the MATLAB figure window. (*if run fast impute is n*)
   4. Figures showing the covariance structure of the TARGET, REFERENCE and ADNI cohorts are displayed.
   5. Figures showing the relationship between the covariance matrices for TARGET vs REFERENCE and TARGET vs ADNI are displayed. These give an indication of the fidelity of the imputation.
   6. Displayed in the Command Window are the relationships between the RAVLT total score and ADNI MEM and, TRAILS B and ADNI EF.
2. Run derive\_ccacc.m
3. On completion of running impute.m then derive\_ccacc.m the CCACC and its component scores are saved in the data directory as TARGET\_CCACC.mat and TARGET\_CCACC.csv.

### 

**5. Top level Functions**

**Impute.m.** This function takes a new TARGET cohort dataset and harmonises it with the REFERENCE dataset. There are three dependent scripts for this function described below (6. Dependent Functions).

1. %% setup
   1. Select Scaled and Merged dataset. Data must be in the correct format, see below (7. Example Data).
   2. Select hold out variable for assessing fidelity of imputation.
2. %% stack TARGET Data
   1. Trims off demographics etc from the cognitive data.
3. %% within sample imputation
   1. Opens parallel processing pool.
   2. Visualises the missingness of the TARGET cohort, yellow bars are collected data blue bars are missing data.
   3. Removes entries where the visit has less than half of the total neuropsychological variables recorded.
   4. Calls knn\_seq\_unique to perform the iterative within cohort imputation that maximises overlap between reference and target data. See below (6. Dependent Functions).
   5. Generates a plot of different RMSE for the varying number of k neighbours and selects the optimal k using the knee point method. This value is then set and the within sample imputation is run.
4. %% stack mega matrix
   1. Loads in the REFERENCE sample of data and appends the TARGET data to create the stacked mega matrix.
5. %% optimise k by holding out ground truth variable for TARGET
   1. Saves then obstructs the hold out variable selected by the user.
   2. Opens a parallel pool to search over 20 values of k to determine optimal value.
   3. Removes the empty rows from the stacked mega matrix.
   4. Uses the REFERENCE data to impute missing values into the TARGET cohort.
      1. Calls knnimpute\_ignore\_nan.m which ensures only k nearest neighbours from the REFERENCE cohort are found when imputing the data. See below (6. Dependent Functions).
   5. Determines the association between the imputed and held out ground truth variable for each iteration of the search across k neighbours.
   6. Calls knee\_pt to select the optimal k using the knee point approach. See below (6. Dependent Functions).
   7. Generates a plot of different RMSE for the varying number of k neighbours and shows the optimal k and RMSE associated with that number of neighbours.
6. %% run imputation using optimal k value and unmasking hidden ground truth variable
   1. Restacks stacked mega matrix.
   2. Removes the empty rows from the stacked mega matrix.
   3. Uses the REFERENCE data to impute missing values into the TARGET cohort.
7. %% fast impute
   1. Stacks REFERENCE and TARGET data and runs sequential imputation (knn\_seq\_unique) from REFERENCE to TARGET using the 3 nearest neighbours.
8. %% store TARGET Data
   1. Binds the demographic information to the imputed TARGET data.
   2. Saves the demographics + imputed TARGET data as a csv as well as a cell with the col headings as a separate variable.
9. %% check association of covariance
   1. Calculates the correlation matrix comparing every cognitive variable for both the TARGET and REFERENCE cohorts.
   2. Loads in the ADNI correlation matrix for reference.
   3. Generates figures for the correlation matrices for the TARGET, REFERENCE and ADNI cohorts. Values are Pearson’s r values with the heatmap bound to +-1.
   4. Extracts the unique values from each correlation matrix (upper triangle without the diagonal).
   5. Plots these unique values and calculates the shared variance (R2) between the TARGET and REFERENCE and, TARGET and ADNI cohorts.
10. %% Relationship with ADNI composites
    1. Calculates and displays the shared variance between RAVLT total and ADNI MEM (imputed) and, TRAILS B and ADNI EF (imputed).
       1. These are good metrics to assess how well the composites were imputed into the TARGET, assuming RAVLT (or CVLT/HVLT) and Trails B were collected in the TARGET cohort.

**derive\_ccacc.m** this function loads the TARGET data and derives the CCACC and its component scores using the imputed and real variables. The CCACC is derived using normalisation (mean and standard deviation) parameters that have been previously calculated based on cognitively normal individuals from the REFERENCE cohort.

1. %% setup
   1. Loads in a look up table for variables within each cognitive domain.
   2. Finds the columns in the TARGET data for each cognitive domain.
2. %% stack domain tests
   1. Stacks variables into domain matrices.
3. %% ensure directions indicate lower is more impaired
   1. Transforms variables so that a lower value indicates more impaired.
4. %% variance normalise individual scores using the REFERENCE sample baseline CN mean, std
   1. Loads in the normalisation parameters derived from the REFERENCE sample.
   2. Normalises each column of each of the domain matrices by the mean and standard deviation of the cognitively normal sample from the REFERENCE cohort.
   3. Sums each of these variance normalised scores to derive the single factor domain score for memory, executive function and general cognition.
5. %% scale then sum domain scores to get ccacc (general is double weighted)
   1. Normalises each domain score by the mean and standard deviation of the cognitively normal sample in the from the REFERENCE cohort.
   2. Sums these values applying double weight to general cognition.
   3. Normalises this sum by the mean and standard deviation from the cognitively normal sample in the REFERENCE cohort.
6. %% stack final data
   1. Append demographics and CCACC, and CCACC domains.
   2. Save Demographics and CCACC scores in TARGET\_CCACC as a .csv along with headings as a .mat file.

**6. Dependent Functions**

**knnimpute\_ignore\_nan.m** this function has a very slight adaptation to the knnimpute.m function from the bioinformatics toolbox, this is on lines 201 to 204 of the function. Here, the function polls through samples until the k closest neighbours *without* a missing variable are selected. Effectively this forces the function to only select neighbours from the REFERENCE cohort and skips neighbours from the TARGET cohort. This is pretty inefficient but does the job fine. This function also returns a vector of the weighted Euclidean distance between the target point and its k nearest neighbours.

**knn\_seq\_unique.m** this function performs the sequential within sample imputation. The aim of this is to group together cognitive visits from the TARGET cohort with the same missing data. Then a stepwise imputation is run ensuring there is maximum overlap between the reference sample within the TARGET cohort (i.e., the sample with full cognitive data) and the target sample (i.e. the groupings of visit with the same missing data), this will maximise the data overlap to determine the k closest neighbours. If the same individual is in the reference and target cohorts, we remove them from the reference cohort. This function is wrapped in a loop to determine the optimal number of k within cohort.

1. %% optimise k
   1. Hide 10% of the hold out variable to determine the fidelity of imputation.
   2. Loop through 1 to 20 neighbours.
   3. Call the stepwise imputation function (nested knn\_seq\_unique) with the optimise k flag down.
      1. %% store temp data matrix
         1. Remove rows that have been nan’d due to less than 50% collected data.
         2. Set flag to poll through the dappled matrix sequentially iterating until all data is full.
      2. %% find the samples with the same missingness
         1. Determine the amount of data missing for each sample.
         2. Find the rows (i.e., the visit) that have the same missingness.
         3. Group the indices of the rows with the same missingness.
         4. Determine which sample(s) have the least missing data and use this as the target (the reference sample is always the full dataset).
         5. Remove these samples from the look up of the indices for the next iteration of the loop.
      3. %% stack the data
         1. Find the subject identifiers from the reference cohort.
         2. Check these against the identifiers from the target sample(s) and remove if overlapping.
         3. Append the target sample(s) to the reference sample.
         4. Impute and save the now complete data as the new reference sample.
      4. %% check missing data, set flag
         1. Check if there are still missing variables.
         2. If there are no longer missing variables (i.e., the matrix has complete variables for each visit) drop the flag to exit the loop.
      5. %% return the matrix
         1. Save the full data back in the original matrix so that the rows are the same as the original data.
   4. Check the association between the 10% of the held-out data and the imputed data for 1 to 20 neighbours.
   5. Use the knee point approach, or, find the minimum RMSE prior to the knee point to select the optimal k (This plot will be messy if there are not many cognitive visits within sample as only 10% of the data is used to determine RMSE (i.e. if 200 visits only RMSE calculated for only 20).
2. Set the optimal k and run through the sequential imputation (i.-v. above) on the full dataset.

**Knee\_pt.m** (Dmitry Kaplan (2022). Knee Point (https://www.mathworks.com/matlabcentral/fileexchange/35094-knee-point), MATLAB Central File Exchange. Retrieved May 17, 2022.).

**7. Example Data**

We have provided a sample of 3 ‘dummy’ cohorts representative of the NIMROD (nim\_dummy\_data.csv), BACS (bacs\_dummy\_data.csv) and AIBL (aibl\_dummy\_data.csv) cohorts. Each cohort has 200 observations (50 dummy participants with 4 years of cognitive data for each). These datasets are relatively small and as such there is more fluctuation in the RMSE plots used to optimise k. As sample size of the TARGET increases the RMSE for different values of k becomes smooth.

Demographic data has been augmented to completely anonymise the dummy data. These 3 examples illustrate how your raw data may look initially, with different variable headings, no scaling, and some irrelevant tests or subtests. For more details on the scaling needed, you can review **Supplementary Table 1** of the associated manuscript**,** which also shows the original/reference test that is used as the scaling denominator.

For each of these dummy cohorts, we will now demonstrate example scripts that scale, rename, and reorder the variables. Outputs will be named ‘example\_dummy\_SCALE\_MERGE’ and exported as both .csv and .mat files, ready to feed into Section 4 above. Feel free to copy relevant scaling/headings to create your own ‘scalemerge\_your\_data.m’ script, to clean and compile your own raw data.

**scalemerge\_nim\_dummy.m.** this function takes the raw NIMROD dummy data (nim\_dummy\_data.csv), cleans and compiles it ready for harmonisation.

1. %% load dummy data and headings
   1. Prompts user to select the csv of the unscaled data.

what is the name of the unscaled data, e.g. nim\_dummy\_data?

* 1. Enter the name of the raw data csv.

what is the name of the unscaled data, e.g. nim\_dummy\_data? nim\_dummy\_data

* 1. Loads the unscaled, raw data as table T.
  2. Loads the headings to which the raw data needs to be mapped (‘headings’).
  3. Creates an empty SCALE\_MERGE cell into which the cleaned and scaled data is loaded, ready for export.

1. %% load demographic data
   1. Phase, RID, sex, date of birth, years of education, age at test and test date are directly loaded. Headings are changed from the original raw titles to the harmonised headings.
   2. Categorical diagnoses are converted to numerically coded diagnoses.
2. %% load cognitive data under new headings, scale as necessary
   1. Where no scaling is needed, data from table T is loaded under the new variable name and new column for the harmonised dataset. Below are the necessary scaling:
      1. INECO backward digitspan (ineco\_dspan) is doubled so that it is now /12 and scaled as per the harmonised dataset.
3. %% catching outliers
   1. This section loads ‘metadata.mat’, which contains the headings of the harmonised dataset, with upper and lower bounds for each variable. Values which fall outside these floor and ceiling values will be removed from the data.
      1. If more than 10% of the data is removed following this process, a warning message will appear. For example, in the NIMROD dummy data, the warning appears for the following variable, prompting you to make sure everything is scaled correctly:
         1. Warning: Greater than 10% of entries removed from socsolvmoves due to data out of range CHECK SCALING! :(
4. %% save your final, clean, re-ordered and scaled data
   1. The SCALE\_MERGE cell array is saved in .mat and .csv format, ready for imputation (Section 4).
      1. NaNs occupy the columns of data which didn’t exist in the original cohort but are present in the harmonised dataset.

**scalemerge\_bacs\_dummy.m.** this function takes the raw BACS dummy data (bacs\_dummy\_data.csv), cleans and compiles it ready for harmonisation. *This dataset shows an example of relatively poor performance of imputation*. This is because this dataset has a.) the lowest overlap of variables between TARGET and REFERENCE, leading to poor RMSE for the hold out variable, and, b.) only cognitively normal subjects, leading to a weaker association between the covariance matrices.

1. %% load dummy data and headings
   1. Prompts user to select the csv of the unscaled data

what is the name of the unscaled data, e.g. bacs\_dummy\_data?

* 1. Enter the name of the raw data csv.

what is the name of the unscaled data, e.g. bacs\_dummy\_data? bacs\_dummy\_data

* 1. Loads the unscaled, raw data as table T.
  2. Loads the headings to which the raw data needs to be mapped (‘headings’).
  3. Creates an empty SCALE\_MERGE cell into which the cleaned and scaled data is loaded, ready for export.

1. %% load demographic data
   1. Phase, RID, sex, years of education, age at test are directly loaded. Headings are changed from the original raw titles to the harmonised headings.
   2. The BACS cohort did not have recorded diagnoses, as all participants were cognitively normal at baseline. Therefore a numerically coded diagnosis of ‘1’ (cognitively normal) is given to each BACS participant.
2. %% load cognitive data under new headings, scale as necessary
   1. Where no scaling is needed, data from table T is loaded under the new variable name and new column for the harmonised dataset. Below are the necessary scaling:
      1. The total FAS letter score is discretised /7.
      2. The animals fluency score is discretised /7.
      3. All CVLT tests and subtests are scaled by a factor of .
      4. WAIS digit symbol task is discretised /10.
      5. Logical verbal recall score scaled to be /25.
      6. WAIS forward and backward digit span tasks scaled to be /12.
3. %% catching outliers
   1. This section loads ‘metadata.mat’, which contains the headings of the harmonised dataset, with upper and lower bounds for each variable. Values which fall outside these floor and ceiling values will be removed from the data.
      1. If more than 10% of the data is removed following this process, a warning message will appear.
         1. Congratulations, the BACS dummy data raises no warnings.
4. %% save your final, clean, re-ordered and scaled data
   1. The SCALE\_MERGE cell array is saved in .mat and .csv format, ready for imputation (Section 4).
      1. NaNs occupy the columns of data which didn’t exist in the original cohort, but are present in the harmonised dataset.

**scalemerge\_aibl\_dummy.m.** this function takes the raw AIBL dummy data (aibl\_dummy\_data.csv), cleans and compiles it ready for harmonisation.

1. %% load dummy data and headings
   1. Prompts user to select the csv of the unscaled data

what is the name of the unscaled data, e.g. aibl\_dummy\_data?

* 1. Enter the name of the raw data csv.

what is the name of the unscaled data, e.g. aibl\_dummy\_data? aibl\_dummy\_data

* 1. Loads the unscaled, raw data as table T.
  2. Loads the headings to which the raw data needs to be mapped (‘headings’).
  3. Creates an empty SCALE\_MERGE cell into which the cleaned and scaled data is loaded, ready for export.

1. %% load demographic data
   1. RID, sex, years of education are directly loaded. Headings are changed from the original raw titles to the harmonised headings.
   2. Phase (cohort identifier) added, diagnosis coded into numerical format, and date of birth converted to datetime format.
2. %% load cognitive data under new headings, scale as necessary
   1. Where no scaling is needed, data from table T is loaded under the new variable name and new column for the harmonised dataset. Below are the necessary scaling/compilations:
      1. MMSE 3-word recognition item-level score is summed.
      2. MMSE serial 7s item-level score is summed.
      3. MMSE word recall item-level score is summed.
      4. FAS letter fluency is discretised /7.
      5. Animals fluency is discretised /7.
      6. MMSE comprehension of instructions item-level score is summed.
      7. MMSE orientation item-level score is summed.
      8. MMSE naming items item-level score is summed.
      9. All CVLT tests and subtests are scaled by a factor of .
      10. Boston Naming Test components are scaled to be /15.
      11. WAIS digit symbol coding task is discretised /10.
      12. WAIS forward and backward digit span tasks scaled to be /12.
3. %% catching outliers
   1. This section loads ‘metadata.mat’, which contains the headings of the harmonised dataset, with upper and lower bounds for each variable. Values which fall outside these floor and ceiling values will be removed from the data.
      1. If more than 10% of the data is removed following this process, a warning message will appear.
         1. Congratulations, the AIBL dummy data raises no warnings.
4. %% save your final, clean, re-ordered and scaled data
   1. The SCALE\_MERGE cell array is saved in .mat and .csv format, ready for imputation (Section 4).
      1. NaNs occupy the columns of data which didn’t exist in the original cohort, but are present in the harmonised dataset.