

# CONTROLLING FOR EFFECTS OF CONFOUNDING VARIABLES ON MACHINE LEARNING PREDICTIONS

Richard Dinga<sup>1,2</sup>, Lianne Schmaal<sup>3</sup>, Brenda W.J.H. Penninx<sup>2</sup>, Dick J. Veltman<sup>2</sup>, Andre F. Marquand<sup>1</sup>

1 Donders Institute, Radboud University, Nijmegen, The Netherlands;

2 Amsterdam UMC, The Netherlands;

3 Orygen, Melbourne, Australia

## ABSTRACT

Machine learning predictive models are being used in neuroimaging to predict information about the task or stimuli or to identify potentially clinically useful biomarkers. However, the predictions can be driven by confounding variables unrelated to the signal of interest, such as scanner effect or motion, limiting the clinical usefulness and interpretation of machine learning models. The most common method to control for confounding effects is regressing out the confounding variables separately from each input variable before machine learning modeling. However, this method is insufficient because machine learning models can learn information from the data that cannot be regressed out. Instead of regressing out confounding effects from each input variable, we propose controlling for confounds post-hoc on the level of machine learning predictions. This allows partitioning of the predictive performance into the performance that can be explained by confounds and performance independent of confounds. This approach is flexible and allows for parametric and non-parametric confound adjustment. We show in real and simulated data that this method correctly controls for confounding effects even when traditional input variable adjustment produces false-positive findings.

## INTRODUCTION

Machine learning predictive models are now being used in clinical neuroimaging research with a promise to be useful for disease diagnosis, predicting prognosis or treatment response (Wolfers et al. 2015). They are also being used in non-clinical settings to find a relationship between biology and personal characteristics such as cognitive capabilities, or identify neural correlates of stimuli or a task (Naseleris et al. 2011). For the correct interpretation of the results and translation of machine learning models into clinical practice, it is important to test that the machine learning predictions are not caused by the effects of confounding variables. For example, in a cognitive experiment, accurate predictions of a stimulus identity can be caused by motion or increased effort due to task difficulty, instead of a neural signal of interest. In a clinical setting, gender, scan-site, motion, or age can cause seemingly accurate machine learning prediction, capturing no other useful information about the disease.

The most common way to control for confounds in neuroimaging is to adjust input variables (i.e., voxels) for confounds using linear regression before subsequent analysis (Snoek et al. 2019). In the case of categorical confounds, this is equivalent to centering each category by its mean, thus the average value of each group with respect to the confounding variable will

be the same. In the case of continuous confounds, the effect on input variables is usually estimated using an ordinary least squares (OLS) regression. The input variables are adjusted by subtracting the estimated effect (i.e., taking the residuals of the confound regression model). This method is, however, problematic for confound adjustment for machine learning models. Since machine learning models are often non-linear, multi-variable, and not fitted using OLS, they can extract information about confounds OLS regression cannot remove. Thus, even after confound adjustment of input variables, the machine learning predictions might still be driven by confounds.

We propose to control for confounds on the level of machine learning predictions instead of the level of input variables. This avoids the problems of input adjustment because we only need to estimate the effect of confounds on the outcome, thus it produces valid results even for complex machine learning models. This method has an intuitive interpretation: estimating the proportion of variance in the outcome can be explained by model predictions that are not already explained by confounding variables.

The remainder of this paper is organized as follows: First, we describe when adjustment of input variables does not sufficiently control for confounding effects. Second, we describe our proposed method, including important caveats relating to the adjustment in regression and classification settings, non-linear and non-parametric adjustment, and cross-validation and permutation schemes. Third, we will show using simulated and real data that the proposed method controls for confounds, even in situations where traditional input adjustment fails. Last, we will discuss how our work relates to other methods, and to which extent can published results be affected by insufficiently adjusted confounds.

## PROBLEMS OF INPUT ADJUSTMENT

Confound adjustment of input variables is the most used method for controlling for confounds in neuroimaging machine learning studies. It is based on a simple idea that if we remove the effects of confounds from the input variables, the results of the subsequent analysis will not be affected by confounds. The effect of confounds is usually estimated using an ordinary least squares (OLS) linear regression for each input variable separately.

$$\mathbf{X} = \mathbf{C}\boldsymbol{\beta}$$

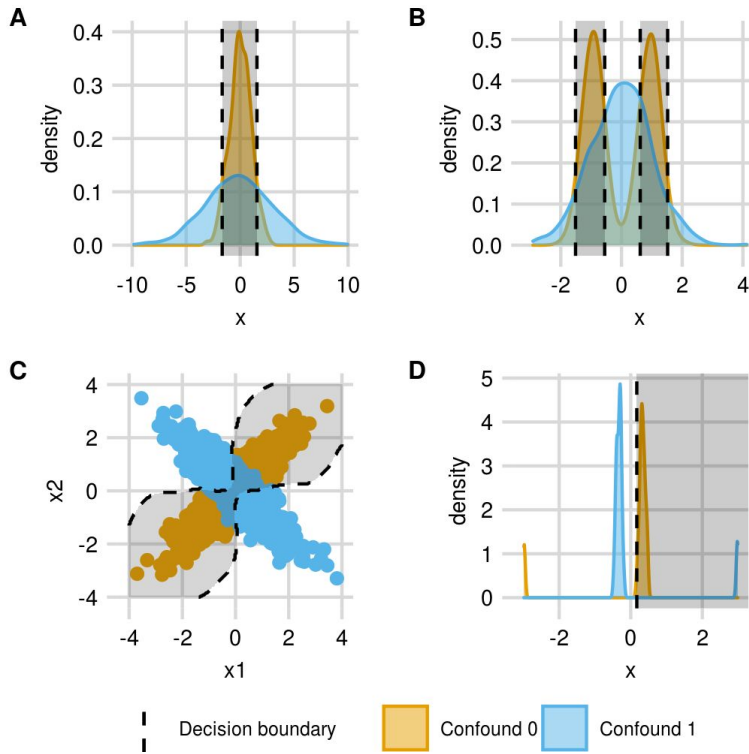
where  $\mathbf{X}$  is a vector of to be adjusted input variables to (i.e., voxel), and  $\mathbf{C}$  is a matrix of confounds, and  $\mathbf{B}$  is a vector of regression coefficients estimated using OLS. The variable is then adjusted by subtracting the estimated effect of confounds

$$\mathbf{X}_{\text{adj}} = \mathbf{X} - \mathbf{C}\boldsymbol{\beta}$$

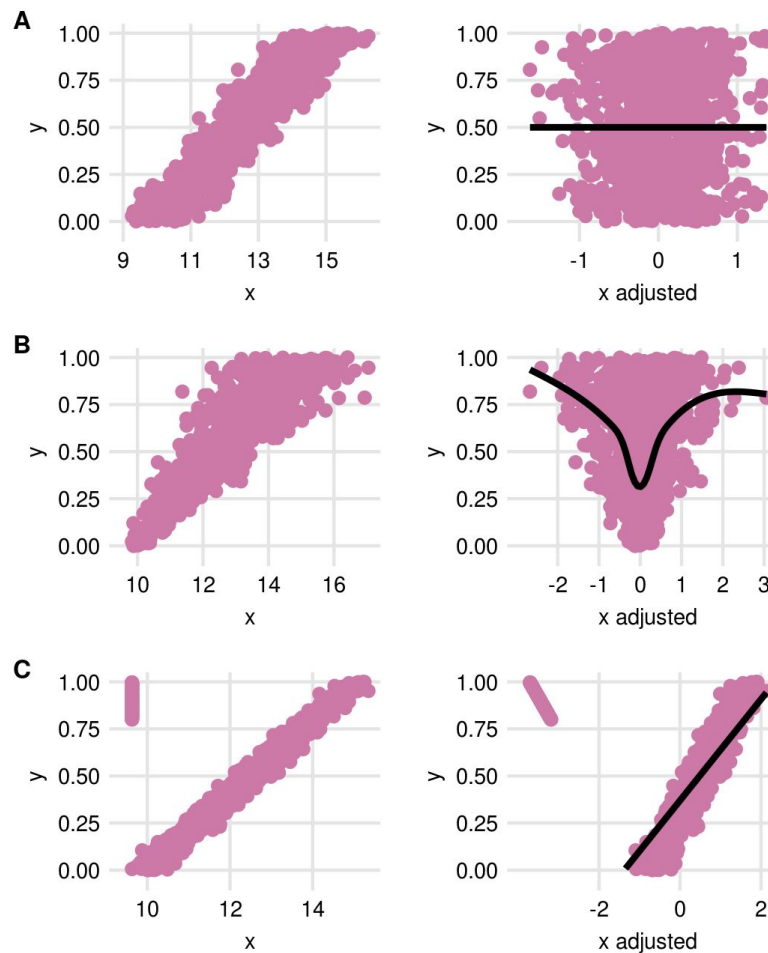
However, this method does not guarantee that the subsequent machine learning analysis will not be affected by confounds. This is because machine learning models can capture information in the data that cannot be captured and removed using OLS. Therefore, even after adjustment, machine learning models can make predictions based on the effects of confounding variables.

There are several sources of confounding information that the OLS adjustment method cannot remove. These are illustrated schematically in Figures 1 and 2 in the context of a machine learning classification and regression, respectively. First, usually, only linear effects are removed, but nonlinear effects will still be present in the data. This can be mitigated by fitting a more complex model using, for example, regressions with polynomial or basis spline expansion. However, even with a complicated model, it is not guaranteed that the model fits the data well. In traditional GLM analysis, this could be easily checked using, for example, diagnostic residual plots. However, such a manual check is not feasible for the large number of variables commonly included in neuroimaging machine learning studies. Second, the confounds can affect scale or shape of the data distribution. For example, in a multi-site analysis, the data variance might be higher in data from one scan-site than another. Therefore, even after centering by site, a nonlinear machine learning model can learn that subjects from one site are more likely to have extreme values of input variables than subjects from the other site (Fig 1A, Fig 2B). This can be mitigated by additional adjusting the scale of the residuals. The simplest way is to divide residuals in each scan site by their standard deviation or model the residuals' standard deviation as a random effect. The modeling approach is performed by COMBAT procedure for adjustment of batch effects of microarray data (Kostro et al. 2014) and scan-site effects of MRI data (Fortin et al. 2017). However, this will not help if the confounds affect not only the scale of the distribution but also its shape, such as skewness or kurtosis (Fig 1b). Third, confounds might have a multivariate effect or they may affect the interaction between input variables. Since each variable is adjusted separately, it is impossible to remove multivariate effects, although they can be easily captured using nonlinear machine learning models (Fig 1C).

The sources of information driven by confounds we reviewed can be used by a nonlinear machine learning models. However, linear machine learning models can also be affected by the effects of confounds even after OLS adjustment. Since OLS regression is fitted to minimize mean squared error, machine learning models that do not minimize mean squared error might still be able to capture confounding information from the data (Fig 1D, Fig 2C). The most prominent example is SVM, which minimizes hinge loss instead of mean squared error. Hinge loss is less affected by outliers than mean squared error, thus if outliers are present in the data OLS will be heavily influenced by these outliers while SVM will still be able to capture information about confounds from the adjusted data.



**Figure 1:** Examples of situations where confound classes are discriminable using machine learning methods after regressing out confounds from input variables. **A:** The confound does not affect the mean, but it affects the variance, which can be learned by a nonlinear machine learning model. **B:** Two groups have the same mean, same variance, but different shape of the distribution, which can also be used by a machine learning model. **C:** The individual variables  $x_1$  and  $x_2$  have the same marginal mean, variance, and shape, however, their joint distribution is different between two groups, which can be used by a nonlinear machine learning model. **D:** Two groups have the same mean due to outliers, however, a robust machine learning model can still discriminate between the two groups.



**Figure 2:** Examples of situations when adjusting input variables for continuous confounding variables works and does not work. Left: original variable before adjustment. Right: variables after confound adjustment. **A:** A situation where confound adjustment works. A linear effect of  $y$  on  $x$  is removed, and there is no more information about  $y$  left in the data. **B:** there is heteroskedastic noise in the input variables. Although the linear effect is removed, there is still information left in the data that can be picked up by a nonlinear machine learning model. **C:** A situation where confound adjustment does not work due to a different loss function. A linear effect is removed from the data as estimated by ordinary least squares regression, which is affected by outliers. A robust machine learning model (in this case support vector machine) can learn a relationship between confound and outcome variable even after adjustment.

## PROPOSED METHOD TO CONTROL FOR EFFECTS OF CONFOUNDING VARIABLES

We propose that the machine learning predictions themselves should be controlled for confounds instead of individual input variables. We treat machine learning predictions as we would any other potential biomarker and apply traditional regression techniques for confound adjustment (multiple linear and logistic regression)(Pourhoseingholi et al. 2012). This approach aims to estimate after the machine learning model is fitted, what proportion of variance is explained by machine learning predictions that cannot be explained by confounds.

Using model predictions as an input to an additional regression model to evaluate its performance is not a new idea; it goes back at least to (Smith and Rose 1995). The proposed approach is closely related to a method known as pre-validation (Tibshirani and Efron 2002; Hoffling and Tibshirani 2008) used in microarray studies to test if a model based on microarray data adds anything to clinical data. In this section, we will first focus on the most general problem of confound adjustment for machine learning regression and machine learning classification in an independent test set. Next, we will describe the usage of this approach when the machine-learning model is evaluated using cross-validation and permutation testing. Last, we will describe non-linear and non-parametric methods for confound adjustment and choice of subjects for creating the adjustment model.

## CONFOUND ADJUSTMENT IN AN INDEPENDENT TEST SET

In a regression setting, there are multiple equivalent ways to estimate the proportion of variance of the outcome explained by machine learning predictions that cannot be explained by the effect of confounds. One is to estimate the partial correlation between model predictions and outcome controlling for the effect of confounding variables.

Given a vector of outcome values  $\mathbf{y}$ , a vector of machine learning predictions  $\mathbf{p}$  and a matrix of confounding variables  $\mathbf{C}$ , the partial correlation is equivalent to the correlation coefficient between residuals of  $\mathbf{p}$  and residuals of  $\mathbf{y}$  after regressing out confounds from both  $\mathbf{p}$  and  $\mathbf{y}$ .

$$\text{pcor}(\mathbf{p}, \mathbf{y} \mid \mathbf{C}) = \text{cor}(\mathbf{p} - \mathbf{C}\boldsymbol{\beta}_p, \mathbf{y} - \mathbf{C}\boldsymbol{\beta}_y)$$

where  $\boldsymbol{\beta}_p$  and  $\boldsymbol{\beta}_y$  are regression coefficients estimated using OLS from a regression equations

$$\begin{aligned}\mathbf{p} &= \mathbf{C}\boldsymbol{\beta}_p + \mathbf{e}_p \\ \mathbf{y} &= \mathbf{C}\boldsymbol{\beta}_y + \mathbf{e}_y\end{aligned}$$

where  $\mathbf{e}_p$  and  $\mathbf{e}_y$  are vectors of error values.

The statistical significance of the partial correlation can be obtained parametrically using a Student's t distribution where the t-statistic is calculated as

$$t(\mathbf{y}, \mathbf{p} \mid \mathbf{C}) = \text{pcor}(\mathbf{y}, \mathbf{p} \mid \mathbf{C}) * \sqrt{(n-2-g)/(1-\text{pcor}(\mathbf{y}, \mathbf{p} \mid \mathbf{C})^2)}$$

is also equivalent to estimating if the estimated effect of model predictions  $\boldsymbol{\beta}_p$  is statistically significant in the multiple linear regression model:

$$\mathbf{y} = \mathbf{p}\boldsymbol{\beta}_p + \mathbf{C}\boldsymbol{\beta}_c$$

or equivalently, this is the same as testing if adding  $\mathbf{p}$  into the model predicting  $\mathbf{y}$  based on confounds will significantly improve  $R^2$  using an F test.

For the interpretation, we can decompose the proportion of explained variance to the variance explained only by confounds ( $\Delta R^2_c$ ), variance explained only by ML predictions ( $\Delta R^2_p$ ), and variance explained by both confounds and ML predictions ( $R^2_{p \cap c}$ ).

$$\Delta R^2_c = R^2_{p+c} - R^2_p$$

$$\Delta R^2_p = R^2_{p+c} - R^2_c$$

$$R^2_{p \cap c} = R^2_{p+c} - \Delta R^2_c - \Delta R^2_p$$

$$R^2_{p+c} = \Delta R^2_p + \Delta R^2_c + R^2_{p \cap c}$$

Where  $R^2_{p+c}$ ,  $R^2_p$ , and  $R^2_c$  are  $R^2$  of models containing ml predictions and confounds, ml predictions, and confounds, respectively.

For categorical outcomes (classification), the logic of the procedure is the same, but instead of performing adjustment using linear regression, we use logistic regression. Our goal is the same as for regression. We want to know what information about the outcome we can explain using model predictions that is not already explained by confounding variables. This can be similarly expressed as the added value of  $\mathbf{p}$  in the model that includes confounds. The optimal way to do this in a logistic model is to evaluate likelihood ratio (LR) or difference in log-likelihoods of two models: 1) predicting outcome using confounding variables

$$\mathbf{M1: logit(y) = C\beta}$$

and 2) predicting the outcome using confounding variables and ML predictions.

$$\mathbf{M2: logit(y) = p\beta_p + C\beta_c}$$

$$\mathbf{LR = likelihood(M1)/likelihood(M2) = loglikelihood(M1) - loglikelihood(M2)}$$

Similarly to the regression setting, this procedure ignores potential miscalibration of predictions, such as systematic overconfidence or underconfidence of estimated probabilities. Thus, we are testing if model predictions contain any information about the outcome that is not already contained in the confounds, but not if the machine learning model's absolute prediction error is better than that of the model using only confounds as predictors. Statistical significance of the partial correlation and likelihood ratio test statistics can be computed parametrically or non-parametrically using a permutation test.

Instead of variance explained, which is not a meaningful measure of model fit for a categorical outcome, we can use a fraction of deviance explained ( $\mathbf{D^2}$ ). This is equivalent to a fraction of variance explained in linear regression, and in logistic regression, it can be interpreted as a proportion uncertainty reduced due to the inclusion of variables to a model (Cameron and Windmeijer 1995). Another benefit of this measure is that it is closely related to the likelihood ratio test that we use to test the added benefit of ML predictions.



$$D^2 = 1 - \text{deviance}(\text{model})/\text{deviance}(\text{null})$$

where deviance is  $2 \times (\text{loglikelihood}(\text{saturated\_model}) - \text{loglikelihood}(\text{model}))$ , the saturated model represents a model with one parameter per observation, and null is an intercept only model.  $D^2$  can be decomposed into a  $\Delta D^2_c$ ,  $\Delta D^2_p$  and  $D^2_{pnc}$  in the same way as  $R^2$  explained before.

## CONFOUND ADJUSTMENT IN CROSS-VALIDATION AND PERMUTATION TEST

The parametric computation of the statistical significance is only valid when the machine learning model is evaluated in an independent test set. However, if a machine learning model is evaluated in cross-validation, traditional parametric tests will produce overly optimistic results. This is because individual errors between cross-validation folds are not independent of each other since when a subject is in a training set, it will affect the errors of the subjects in the test set. Thus a parametric null-distribution assuming independence between samples will be too narrow and therefore producing overly optimistic p-values. The recommended approach to test the statistical significance of predictions in a cross-validation setting is to use a permutation test (Golland and Fischl 2003; Noirhomme et al. 2014). The outcome values are randomly permuted many times, and for each permutation, the cross-validation is performed using the permuted outcome values instead of original outcome values. P-value is then calculated as a proportion of cross-validation results performed using the permuted data that is better than cross-validation results obtained using the original, non-permuted data.

A similar permutation testing procedure can also be used to obtain a null-distribution of a confound adjusted test statistic, the partial correlation for regression, and the likelihood ratio for classification as described above. An important caveat is that the permutation procedure should only affect the relationship between input variables and the outcome, but not the relationship between the outcome and confounding variables (Fig 3). The permutation needs to be performed on the rows of the input variables but not on the outcome labels and not on the confounding variables. If only the outcomes were shuffled, the results would be biased because the confounds will no longer be related to the outcomes, and thus this will not create a correct null distribution.



Original data			Wrong shuffle			Correct shuffle		
5	5	5	5	5	2	5	2	2
4	4	4	4	4	3	4	3	3
3	3	3	3	3	5	3	5	5
2	2	2	2	2	1	2	1	1
1	1	1	1	1	4	1	4	4
x	c	y	x	c	y	x	c	y

*Figure 3: valid and invalid permutation scheme in the presence of confounds. Given input variables  $x$ , confounds  $c$ , and outcome values  $y$ , the incorrect way is to shuffle only  $y$ , which would remove the relationship between  $x$  and  $y$  but also between  $c$  and  $y$ , leading to biased results. The correct way is to remove the relationship between  $x$  and  $y$  but keep the relationship between  $c$  and  $y$  fixed.*

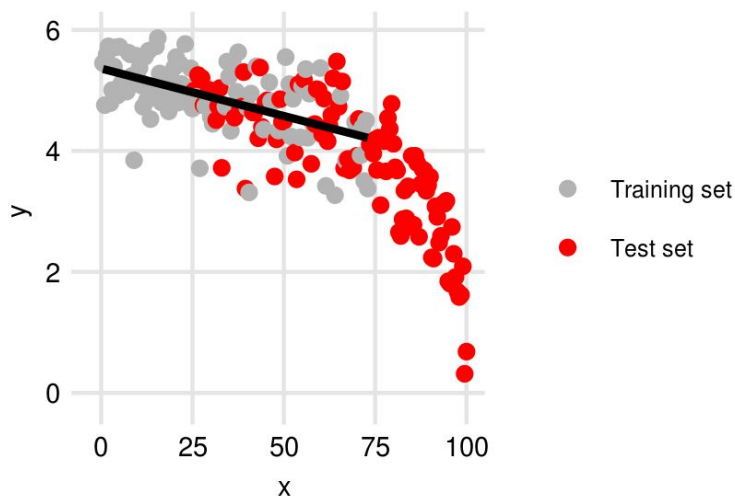
## WHICH DATA TO USE TO CREATE THE ADJUSTMENT MODEL

The model used to perform confound adjustment can be estimated using all available data, however, in some cases, it has been recommended in the literature to use only a subset of the data to fit the confound adjustment model. (Dukart et al. 2011) recommended performing confound adjustment only based on the data from healthy controls but omit data from cases, with the justification that otherwise we might be removing the effect of the disease as well, due to its interaction with the confound. However, as was pointed out by (Linn et al. 2016), this procedure will not sufficiently remove the effects of confounds, and thus it will produce biased results as illustrated in Figure 4. This is because data from healthy controls are insufficient to estimate the effect of confounds in subjects with a disease.

(Snoek et al. 2019) recommend performing confound adjustment only based on the data from the training set but omit the test set to avoid a negative bias that can even lead to a significant below chance performance. If an effect of a variable on the outcome in the whole dataset is 0, then the effect learned in the training set will have an opposite sign in the test set, leading to negatively biased results.

To avoid this problem, (Snoek et al. 2019) recommend that the confound adjustment of the test set data is performed using a model developed only on the data in the training sets, thus no artificial dependency between the training and test set data is created. However, in some situations, this is not possible because the levels of confounds we want to adjust for are not presented in the training data, for example, in the case of confounding effects of scan-sites

where the test set consists of scan-sites that are not present in the training set. A second problem relates to a similar issue when adjusting only based on the data from healthy controls: if there is a difference in confound distribution between training and test set, then the test set will be insufficiently adjusted, and thus the results might be confounded. Since our proposed method is based on adjusting model predictions, it will not create an artificial dependency between the training and test set data, thus it will not lead to artificially lower than chance predictions.



**Figure 4:** Adjustment of the test set based on training set data, or patients based on healthy controls data might be insufficient and thus should be avoided.

## WHAT VARIABLES TO ADJUST FOR

Another important consideration is what variables we should adjust for. Since the goal is to evaluate machine learning predictive models, this is not necessarily a causal question, but mostly a practical one and thus entirely problem-dependent. We might want to ask the two questions: 1) Is the machine learning model predicting a signal of interest or signals that are not of interest (e.g., scan-site, age, motion, BMI)? 2) Does the machine-learning model predict the actual outcome or perhaps something correlated to the outcome? For example, if we want to create a predictive model of future disease remission based on MRI data, we may want to ask whether our model based on MRI data predicts future remission or just current disease severity (which may be highly related to future remission). Therefore, we might want to control for current severity to estimate the added value of MRI data for prediction. However, maybe the goal of our study is to create an MRI based predictive model to replace a clinical assessment. In that case, the current severity of symptoms would not be considered a confound to control for, although we might still want to evaluate if the variance explained by severity is the same as the variance explained by the MRI based machine learning model.

## NONLINEAR AND NONPARAMETRIC ADJUSTMENT

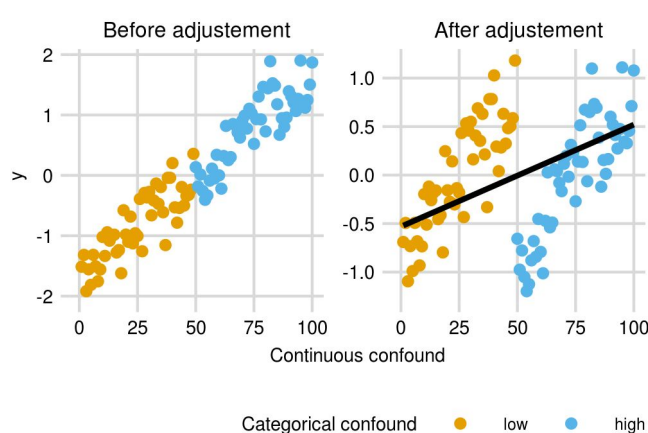
The effect of confounds might not be strictly linear, thus it is important to be able to accommodate also nonlinear effects. If this is not considered, the residual effects of confounds can still be present in the data and can bias results (Figure 5). Since our proposed method is just an application of traditional linear or logistic regression, the same methods for estimating nonlinear effects apply. One way is to expand confounding variables using a polynomial or basis spline expansions. Another possibility is to estimate the effects of confounding variables through non-parametrically generalized additive models and smoothing splines (Hastie and Tibshirani 1987; Wood 2017), according to the following equation:

$$y = p\beta_p + f(C)$$

$$\text{logit}(y) = p\beta_p + f(C)$$

Where  $f(C)$  is a parametric or non-parametric smoothing function.

The somewhat common, but invalid approach to combat the nonlinear effects of confounds is categorizing confounding variables. For example, instead of correcting for BMI, the correction is performed for categories of low, medium, and high BMI. Such a categorization is unsatisfactory because it keeps residual confounding within-category variance in the data, which can lead to both false positive and false negative results (Austin and Brunner 2004). False-positive results because there can still be residual confounding information presented in the input data, and false negative because the variance in the data due to confounding variables will lower the statistical power of a test. Thus categorizing continuous confounding variables should not be performed. Instead, other parametric or nonparametric approaches for the modeling of nonlinear effects should be used.



**Figure 5:** Categorizing continuous confound variable before adjustment might lead to insufficiently adjusted data, with the residual confounding signal still present in the data.

## EXAMPLE ANALYSIS

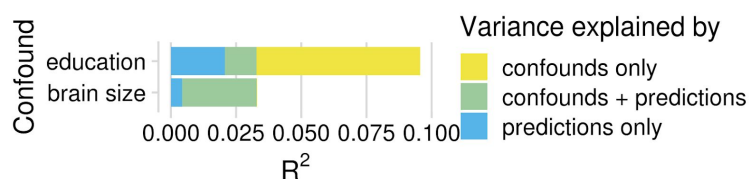
To illustrate the usage of the proposed approach, we performed an example analysis in a similar way that can be performed in practice. We aimed to predict a fluid intelligence (FI)

score based on brain region of interest volumetric data. We performed a separate control for two confounding variables, 1) brain size, to evaluate if a machine learning model learned any useful patterns from the data, above and beyond what can be explained by brain size. 2) length of education, to evaluate an added benefit of a brain-based machine learning model to a variable that is much easier to obtain and test if the ML model learned to predict anything else than function of education length.

**Dataset:** We used the UK Biobank region of interest volume data as obtained from the structural MRI images using FSL FAST. The details about the dataset, preprocessing, and volume extraction procedure can be found elsewhere (Miller et al. 2016; Alfaro-Almagro et al. 2018). From the dataset of size  $n=21407$  we excluded 2149 subjects due to missing FI data. Next, we split the data into halves while preserving the FI distributions between a training and test set. We also excluded 4461 subjects from the test set due to missing education data, leaving the size of the training set at 9630 and the test set 5167.

**Method:** We fitted a ridge regression model implemented in the glmnet package (Friedman et al. 2010) in the training set to predict FI. We evaluated the predictions in the test set, by including them in a regression model together with the brain size variable and in another regression model with an education length variable. To take into account nonlinear effects of education, we used cubic spline expansion with 5 knots. This procedure allowed us to estimate the proportion of variance of the FI, explained by confounding variables and a proportion of FI variance explained by predictions alone, thus effectively controlling the results for the effects of confounding variables.

**Results:** Machine learning predictions explained a small proportion of FI variance ( $R^2=0.03$ ,  $p < 0.001$ ), which could be mostly but not completely explained by brain size ( $R^2_{\text{shared}} = 0.025$ ,  $\Delta R^2_{\text{MLpredictions}}=0.005$ ). This indicates that ML can learn patterns related to FI above and beyond brain size, even though the added benefit is small. Brain size was not able to explain additional variance not already explained by ML predictions ( $\Delta R^2_{\text{brain size}} = 0$ ), which was expected since the effect of brain size can be implicitly learned from the individual ROI volumes. The  $R^2$  due to ML predictions ( $R^2_{\text{ml\_predictions}} = 0.025$ ) was smaller than  $R^2$  due to education length ( $R^2_{\text{education}}=0.08$ ). However, the ML predictions explained a proportion of variance that could not have been explained by the effect of education. Therefore the effect of ML predictions was still highly statistically significant.



**Figure 6:** Results of machine learning prediction of fluid intelligence based on brain imaging data, taking into account the confounding effects of education length or brain size. Machine learning predictions were able to predict the proportion of variance not already explained by the impact of confounding variables, therefore the results were not fully driven by confounds.

## EXPERIMENTS

For the empirical validation of the proposed method, we simulated confounded datasets for regression and classification and also used a real dataset with the artificially created confounded outcome variable. The dataset for regression was simulated similarly to the dataset in figure 2B. The confounding variable was uniformly distributed between 0 and 100. Input variable  $x$  was simulated as a function of the confound with heteroscedastic noise term according to:

$$x = 10 + 5 \cdot \text{confound} + \text{norm}(\text{mean}=0, \text{sd}=1) \cdot (10 + \text{confound} \cdot \text{weight})$$

where the weight was set to 0.1, 0.2, and 0.3 representing low, medium, and high effect of confounding. The outcome variable  $y$  was simulated as a function of the confounding variable according to:

$$y = 100 + 0.2 \cdot \text{confound} + \text{norm}(\text{mean}=0, \text{sd}=5)$$

therefore, there was no relationship between  $x$  and  $y$  that cannot be explained by the effect of the confounding variable.

The dataset for classification was simulated similarly to figure 1C. First, a categorical confounding variable with two categories was created. Next, two input variables  $x_1$  and  $x_2$  were sampled from a bivariate normal distribution with mean = 0, 0 and covariance matrix =  $\begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$  or  $\begin{bmatrix} 1 & -0.9 \\ -0.9 & 1 \end{bmatrix}$ , depending on the category of the confounding variable. Outcome variable  $y$  was created as a function of the confounding variable, such as the probability of  $y=1$  was 0.6, 0.7, 0.8 representing low, medium, and high confounding, if the confounding variable was 0.

We use support vector regression with a radial basis function (RBF) kernel with  $C=1$  for the regression problem (Smola and Schölkopf 2004) and Gaussian process classification (Rasmussen and Williams 2005) with an automatic sigma estimation for the classification problem, both implemented in kernlab (Karatzoglou et al. 2004). For each simulation, we created a dataset of size 200. This dataset was split into halves used as a training set and a validation set where the statistical significance was assessed using a parametric test. We also performed a 5 fold cross-validation where the statistical significance was assessed using a permutation test. Our goal is to adjust for the confounding variables. We perform two types of confound adjustments. 1: adjustment of input variables according to (Snoek et al. 2019), where the training set data and test set data were first adjusted for confounds by regressing out the effect of confounding variables estimated using the training set data. 2: additional adjustment of the ML predictions according to the proposed method using linear regression for the regression problem and logistic regression for the classification problem. The statistical significance was assessed either parametrically on the holdout set, or nonparametrically using a permutation test for cross-validation. Since the datasets were created, such as there is no relationship between input variables and the outcome that is not

explained by the effect of confounding variables, the valid confound adjustment method should produce statistically significant results with  $p < 0.05$  around 5% of the times.

We also used a real neuroimaging dataset consisting of region of interest (ROI) thickness data from the ABIDE dataset (Di Martino et al. 2014) obtained from the preprocessed connectome project (Craddock et al. 2013). The dataset consisted of 97 regions of interest (ROI), including cortical ROI obtained using the ANTs pipeline according to Desikan-Killiany protocol (Klein and Tourville 2012) and FreeSurfer white matter and non-cortical ROI (Fischl 2012)(See <https://mindboggle.readthedocs.io/en/latest/labels.html> for a complete list of ROI). We used data from 530 healthy control subjects, with age at scan spanning between 6 to 56 years (median = 14.66, IQR=8.2).

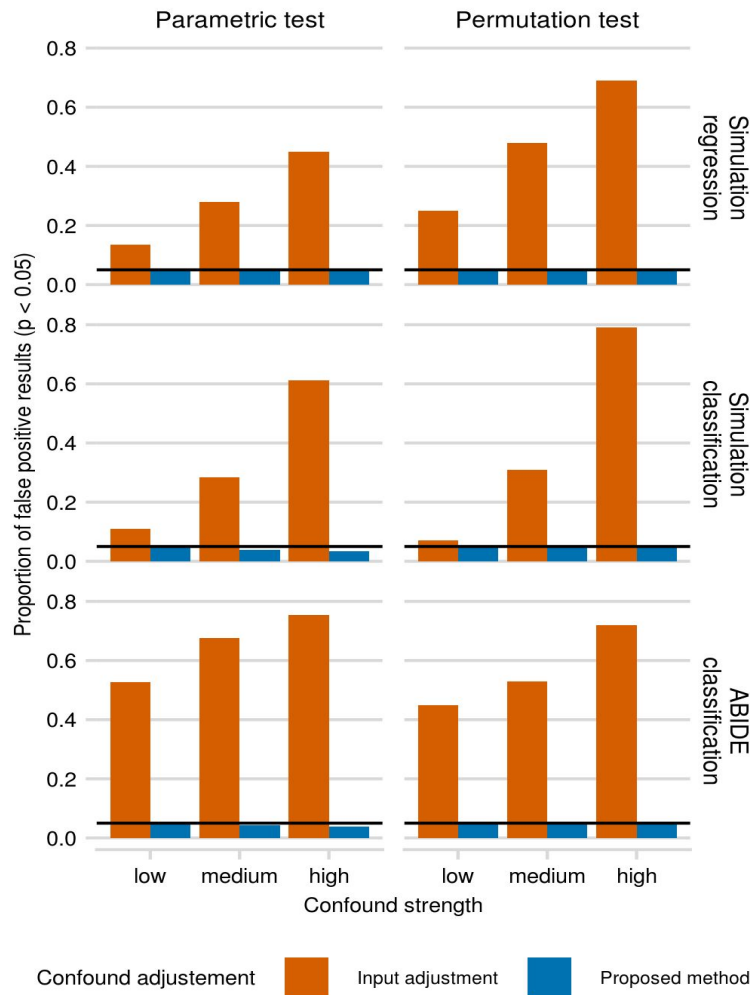
For each simulation, we randomly created a categorical outcome variable  $y$  confounded by age, where the probability of  $y=1$  was based on subjects age according to:

$$\log [p(y=1)/p(y=0)] = (\text{age} - 14.66) * \text{weight}$$

where weight was set to 3, 4, and 5 representing low, medium, and high confounding, since the outcome variable was created solely as a function of age, there should be no signal in the data after adjustment for age.

## RESULTS

Since the datasets were created so that there is no signal in the data that cannot be explained by the confounding variables, all statistically significant results are false positive. Therefore, the successful deconfounding method should have a false positive rate at  $p < 0.05$  around 5%. In simulated regression and classification datasets, the input adjustment method did not sufficiently control for confounds, and significantly above chance performance was obtained. The percentage of false-positive results could be made arbitrarily high (even as high as 100%), depending on the amount of confounding. In the real dataset, the false positive rate as high as 72% was observed when predicting the outcome that was created as a function of age after correcting for age. In all tested scenarios, the proposed confound adjustment method of the machine learning predictions achieved a false positive rate of around 5%. This was the case when the testing was performed on the independent hold-out set using a parametric test or in cross-validation using a permutation test (Figure 6).



**Figure 6:** results of traditional input adjustment method and proposed output adjustment method. The datasets were created, such as there is no signal in the data that cannot be explained by the effect of confounding variables, therefore after successful confound adjustment, the proportion of statistically significant results ( $p < 0.05$ ) should be around 5% (black line). We simulated confounded data for regression and classification problems. Also, We used a real neuroimaging dataset (ABIDE), where the confounding variable was age, and we created an artificial categorical outcome as a function of age. The amount of confounding effect was varied. The statistical significance was assessed using a parametric test in the hold-out set, or a permutation test in 5-fold cross-validation.

## DISCUSSION

Controlling for confounds at the level of model predictions successfully managed for confounds under low, medium, and high confounding, even in situations where the traditional input variable adjustment failed. This approach has a simple and intuitive interpretation: what proportion of variance in the outcome can be explained by model predictions that cannot be explained by confounding variables? Or in other words, does predict the outcome using confounding variables improve when adding model predictions? This approach evaluates if



the model predictions have any information about the outcome that is not already present in the confounding variables, thus it is insensitive to model miscalibration.

Traditional input variable adjustment failed to sufficiently control for confounds in simulated and real datasets. This is because input variable adjustment can not remove all effects that can be learned by machine learning methods. This includes cross-validated input adjustment as proposed by (Snoek et al. 2019) and adjustment using a location and scale adjustment model as used in ComBat (Fortin et al. 2017). Therefore, it is possible that some of the previously published machine learning results are due to insufficiently adjusted confounding but not the signal of interest. Machine learning methods vulnerable to this problem include all non-linear machine learning methods and linear machine learning methods that are fitted optimizing a different function than a regression used for input adjustment, such as support vector machines. Support vector machines optimize a hinge loss, which is more robust to extreme values than a squared loss used for input adjustment. Therefore, the presence of outliers in the data will lead to improper input adjustment that can be exploited by SVM. Studies using penalized linear or logistic regression (i.e., lasso, ridge, elastic-net) and the linear Gaussian process should not be affected by these confounds (assuming they are sufficiently adjusted) since these models are not more robust to outliers than OLS regression.

The proposed approach is closely related “pre-validation” method used in microarray studies to test if a model based on micro-array data adds something to clinical predictors (Tibshirani and Efron 2002; Hoffling and Tibshirani 2008). Here we argued that this approach can also be used to control for confounds of machine learning predictions in general and suggest using  $D^2$  and  $R^2$  and their decompositions to interpret the results. Multiple alternative approaches for controlling confounding effects exist and can be used in a machine learning setting. One possibility is to use a permutation test where the permutations are performed within the confound groups (Winkler et al. 2015). For example, if we wish to control for effects of scan sites, labels would be shuffled within each scan site separately. Thus if a model’s performance is driven by the scan site effects, this will be reflected in the permutation-based null-distribution. This method can be used even with a complex dependence structure of confounding variables (Winkler et al. 2015), however, it is limited only to categorical confounds, and the number of confounding variables limits the number of possible permutations.

A related approach to the proposed method, sometimes used in neuroimaging, is to regress out confounds from the target variables and apply machine learning to predict these residuals. This approach is similar to testing for partial correlations, as we proposed, with few critical distinctions. First, it cannot be applied to classification problems. Second, to avoid data leakage, the confound regression have to be based on the training data (Snoek et al. 2018), which, as we argued, is not always possible. Third, since the effect of confounds should not be estimated based on the training set data, they might not be optimally estimated, leaving residual confounding effect in the data leading to insufficient adjustment. Fourth, the interpretation is limited, since it only allows for testing the partial effects of ML predictions, while the proposed method allows for the estimation of variance that can be

explained by both confounds and machine learning predictions, confounds only, and machine learning predictions only.

Another possibility is to use balancing or matching of subjects according to levels of their confounding variables (Pourhoseingholi et al. 2012; Rao et al. 2017; Chyzhyk et al. 2018), to create a dataset where the confounding variable is not related to the outcome. Since only a subset of available subjects is used, this leads to data loss and highly variable estimates. To combat this, sometimes subjects are not selected but weighted according to the outcome model's model based on confounding variables. Thus all subjects are still used to estimate the model performance, but their influence on the final results is different. These methods are problematic because they change the distribution of variables in the test set, which no longer matches the distribution of the original dataset or the population. For example, controlling for a sex effect in the machine learning prediction of Autism diagnosis resampling methods would be interpreted as a performance of the machine learning model in the population where sex is not related to the autism diagnosis. Since more boys are diagnosed with autism than girls, resampling methods will put a higher weight on predictions for girls, which might lead to a selection of a model that makes worse predictions in boys even though the majority of autism diagnoses is in boys.

A related problem of this method is that it cannot be used to evaluate the additional benefit of the machine learning model predictions over what can be already predicted by confounds since by construction confounding variables have no predictive power in the newly created population. It might be tempting to say that the model's added value equals the performance of the model in this newly created population. This is, however, not the case. As shown by (Pepe et al. 2012) and (Janes and Pepe 2008), this can severely underestimate and also overestimate the added value and even change ranks of competing models. This has serious implications. It can lead to selecting the worse model for prediction, missing potentially important biomarker, or selecting apparently strong biomarker that, in reality, does not add much to what can be already predicted using confounds.

## CONCLUSION

We showed that confound adjustment of input variables could fail to adequately control for confounding effects when machine learning methods are used. For this reason, we propose that confound adjustment of input variables should be avoided, and the already published machine learning studies employing it should be interpreted with care. We presented a simple approach of controlling for confounds of the machine learning predictions themselves. This approach produced valid results even under heavy and complicated confounding.

- Alfaro-Almagro, F., Jenkinson, M., Bangerter, N.K., Andersson, J.L.R., Griffanti, L., Douaud, G., Sotiropoulos, S.N., Jbabdi, S., Hernandez-Fernandez, M., Vallee, E., Vidaurre, D., Webster, M., McCarthy, P., Rorden, C., Daducci, A., Alexander, D.C., Zhang, H., Dragonu, I., Matthews, P.M., Miller, K.L., Smith, S.M., 2018. Image processing and Quality Control for the first 10,000 brain imaging datasets from UK Biobank. *Neuroimage* 166, 400–424. <https://doi.org/10.1016/j.neuroimage.2017.10.034>
- Austin PC, Brunner LJ (2004) Inflation of the type I error rate when a continuous confounding variable is categorized in logistic regression analyses. *Stat Med* 23:1159–1178. <https://doi.org/10.1002/sim.1687>
- Chyzhyk D, Varoquaux G, Thirion B, Milham M (2018) Controlling a confound in predictive models with a test set minimizing its effect. In: 2018 International Workshop on Pattern Recognition in Neuroimaging, PRNI 2018. IEEE, pp 1–4
- Craddock C, Benhajali Y, Chu C, et al (2013) The Neuro Bureau Preprocessing Initiative: open sharing of preprocessed neuroimaging data and derivatives. *Front Neuroinform* 7:. <https://doi.org/10.3389/conf.fninf.2013.09.00041>
- Di Martino A, Yan C-G, Li Q, et al (2014) The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Mol Psychiatry* 19:659–67. <https://doi.org/10.1038/mp.2013.78>
- Dukart J, Schroeter ML, Mueller K, Alzheimer's Disease Neuroimaging Initiative TADN (2011) Age correction in dementia--matching to a healthy brain. *PLoS One* 6:e22193. <https://doi.org/10.1371/journal.pone.0022193>
- Fischl B (2012) FreeSurfer. *Neuroimage* 62:774–781
- Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. Stat. Softw.* 33, 1–22.
- Fortin J-P, Parker D, Tunç B, et al (2017) Harmonization of multi-site diffusion tensor imaging data. *Neuroimage* 161:149–170. <https://doi.org/10.1016/j.neuroimage.2017.08.047>
- Golland P, Fischl B (2003) Permutation tests for classification: Towards statistical significance in image-based studies. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 2732:330–341. [https://doi.org/10.1007/978-3-540-45087-0\\_28](https://doi.org/10.1007/978-3-540-45087-0_28)
- Hastie T, Tibshirani R (1987) Non-Parametric Logistic and Proportional Odds Regression. *Appl Stat* 36:260. <https://doi.org/10.2307/2347785>
- Höfling, H., Tibshirani, R., 2008. A study of pre-validation. *Ann. Appl. Stat.* 2, 643–664. <https://doi.org/10.1214/07-AOAS152>

- Janes H, Pepe MS (2008) Matching in Studies of Classification Accuracy: Implications for Analysis, Efficiency, and Assessment of Incremental Value. *Biometrics* 64:1–9. <https://doi.org/10.1111/j.1541-0420.2007.00823.x>
- Karatzoglou A, Smola A, Hornik K, Zeileis A (2004) kernlab - An S4 Package for Kernel Methods in R. *J Stat Softw* 11:1–20. <https://doi.org/10.18637/jss.v011.i09>
- Klein A, Tourville J (2012) 101 Labeled Brain Images and a Consistent Human Cortical Labeling Protocol. *Front Neurosci* 6:171. <https://doi.org/10.3389/fnins.2012.00171>
- Kostro D, Abdulkadir A, Durr A, et al (2014) Correction of inter-scanner and within-subject variance in structural MRI based automated diagnosing. *Neuroimage* 98:405–415. <https://doi.org/10.1016/j.neuroimage.2014.04.057>
- Linn KA, Gaonkar B, Doshi J, et al (2016) Addressing Confounding in Predictive Models with an Application to Neuroimaging. *Int J Biostat* 12:31–44. <https://doi.org/10.1515/ijb-2015-0030>
- Miller, K.L., Alfaro-Almagro, F., Bangerter, N.K., Thomas, D.L., Yacoub, E., Xu, J., Bartsch, A.J., Jbabdi, S., Sotiropoulos, S.N., Andersson, J.L.R., Griffanti, L., Douaud, G., Okell, T.W., Weale, P., Dragonu, I., Garratt, S., Hudson, S., Collins, R., Jenkinson, M., Matthews, P.M., Smith, S.M., 2016. Multimodal population brain imaging in the UK Biobank prospective epidemiological study. *Nat. Neurosci.* 19, 1523–1536. <https://doi.org/10.1038/nn.4393>
- Naselaris T, Kay KN, Nishimoto S, Gallant JL (2011) Encoding and decoding in fMRI. *Neuroimage* 56:400–10. <https://doi.org/10.1016/j.neuroimage.2010.07.073>
- Noirhomme Q, Lesenfants D, Gomez F, et al (2014) Biased binomial assessment of cross-validated estimation of classification accuracies illustrated in diagnosis predictions. *NeuroImage Clin* 4:687–694. <https://doi.org/10.1016/J.NICL.2014.04.004>
- Pepe MS, Fan J, Seymour CW, et al (2012) Biases Introduced by Choosing Controls to Match Risk Factors of Cases in Biomarker Research. *Clin Chem* 58:1242. <https://doi.org/10.1373/CLINCHEM.2012.186007>
- Pourhoseingholi MA, Baghestani AR, Vahedi M (2012) How to control confounding effects by statistical analysis. *Gastroenterol Hepatol from bed to bench* 5:79–83
- Rao A, Monteiro JM, Mourao-Miranda J, Alzheimer's Disease Initiative AD (2017) Predictive modelling using neuroimaging data in the presence of confounds. *Neuroimage* 150:23–49. <https://doi.org/10.1016/j.neuroimage.2017.01.066>
- Rasmussen CE, Williams CKI (2005) Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series). The MIT Press
- Smith, E.P., Rose, K.A., 1995. Model goodness-of-fit analysis using regression and related techniques. *Ecol. Modell.* 77, 49–64. [https://doi.org/10.1016/0304-3800\(93\)E0074-D](https://doi.org/10.1016/0304-3800(93)E0074-D)
- Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. *Stat. Comput.* 14:199–222

- Snoek L, Miletic S, Scholte HS (2019) How to control for confounds in decoding analyses of neuroimaging data. *Neuroimage* 184:741–760.  
<https://doi.org/10.1016/J.NEUROIMAGE.2018.09.074>
- Tibshirani, R.J., Efron, B., 2005. Pre-validation and inference in microarrays. *Stat. Appl. Genet. Mol. Biol.* 1. <https://doi.org/10.2202/1544-6115.1000>
- Winkler AM, Webster MA, Vidaurre D, et al (2015) Multi-level block permutation. *Neuroimage* 123:253–268. <https://doi.org/10.1016/j.neuroimage.2015.05.092>
- Wolfers T, Buitelaar JK, Beckmann C, et al (2015) From estimating activation locality to predicting disorder: a review of pattern recognition for neuroimaging-based psychiatric diagnostics. *Neurosci Biobehav Rev.* <https://doi.org/10.1016/j.neubiorev.2015.08.001>
- Wood SN (2017) Generalized additive models: An introduction with R, second edition