Pitfalls and Remedies for Cross Validation with Multi-trait Genomic Prediction Methods

Daniel Runcie[*,1] **and Hao Cheng**[†]
[*]Department of Plant Sciences, University of California Davis, Davis, CA, USA, [†]Department of Animal Science, University of California Davis, Davis, CA, USA

ABSTRACT Incorporating measurements on correlated traits into genomic prediction models can increase prediction accuracy and selection gain. However, multi-trait genomic prediction models are complex and prone to overfitting which may result in a loss of prediction accuracy relative to single-trait genomic prediction. Cross-validation is considered the gold standard method for selecting and tuning models for genomic prediction in both plant and animal breeding. When used appropriately, cross-validation gives an accurate estimate of the prediction accuracy of a genomic prediction model, and can effectively choose among disparate models based on their expected performance in real data. However, we show that a naive cross-validation strategy applied to the multi-trait prediction problem can be severely biased and lead to sub-optimal choices between single and multi-trait models when both the correlated "secondary" traits and the focal trait are measured on the same individuals. We use simulations to demonstrate the extent of the problem and propose two partial solutions: a semi-parametric method for correcting the cross-validation estimates of prediction accuracy, and a modified cross-validation approach which we call CV2*: validating model predictions against focal trait measurements from genetically related individuals. The current excitement over high-throughput phenotyping suggests that more comprehensive phenotype measurements will be useful for accelerating breeding programs. Using an appropriate cross-validation strategy should more reliably determine if and when combining information across multiple traits is useful.

KEYWORDS
Cross Validation
Genomic Prediction
Linear Mixed Model
multi-trait

INTRODUCTION

Genomic Selection (GS) aims to increase the speed and accuracy of selection in breeding programs by predicting the genetic worth of candidate individuals or lines earlier in the selection process, or for individuals that cannot be directly phenotyped (Meuwissen *et al.* 2001; Hayes *et al.* 2009; Crossa *et al.* 2017). Genomic selection works by training statistical or Machine Learning models on a set of completely phenotyped and genotyped individuals, and then using the trained model to predict the genetic worth of unmeasured individuals. If the predictions are reasonably accurate, selection intensity can be increased because the population size of candidate individuals is larger or their true genetic worth is estimated more accurately.

Predictions of genetic values are usually based only on the genotypes of the new individuals, or on their pedigrees. However predictions can in some cases be improved by including measurements of "secondary" traits that may not be of direct interest but are easier or faster to measure (Thompson and Meyer 1986; Pszczola *et al.* 2013; Lado *et al.* 2018). This is one goal of multi-trait genomic prediction. Multi-trait prediction is most useful for increasing the accuracy of selection on a single focal trait when that trait has low heritability, the "secondary" trait(s) has high heritability, and the genetic and non-genetic correlations between the traits are large and opposing (Thompson and Meyer 1986; Jia and Jannink 2012). With the advent of cheap high-throughput phenotyping, there is great interest in using measurements of early-life or easily accessible traits to improve prediction of later-life or more expensive traits, and multi-trait prediction models are attractive methods for leveraging this information (Pszczola *et al.* 2013; Rutkoski *et al.* 2016; Fernandes *et al.* 2017; Lado *et al.* 2018).

A large number of genomic prediction methods are available, and the best model varies across systems and traits (Heslot *et al.*

Manuscript compiled: Sunday 31st March, 2019
[1]Corresponding author: One Shield Ave, Davis, CA 95616 deruncie@ucdavis.edu

2012; de Los Campos *et al.* 2013). Most models involve some from of multiple linear regression and can be written as a linear mixed effect model (Gianola 2013). Due to their complexity and often high-dimensional nature, genomic prediction methods are prone to overfitting and require regularization to perform well on new data. Therefore, comparing models based on their ability to fit existing data (ex. with R^2) is unreliable; every candidate model could explain 100% of the variation in a typical-size dataset.

Instead, prediction models are generally compared by cross-validation (Meuwissen *et al.* 2001; Utz *et al.* 2000; Gianola and Schon 2016). The basic idea of cross-validation is to separate the model fitting and tuning process from the model evaluation process by using separate datasets for each (Friedman *et al.* 2001). This penalizes models that fit too closely to one data set at the expense of generalization. In this way, cross-validation is meant to accurately simulate the real-world usage of the model: predicting the genetic values of un-phenotyped individuals; i.e. those not available during the model fitting process itself. Rather than requiring new data *per se*, cross-validation works by splitting an existing dataset into non-overlapping "training" and "testing" partitions, fitting the candidate model to the former, and then evaluating it on its accuracy at predicting the latter. Common measures of accuracy include Pearson's ρ or the square root of the average squared error (RMSE) Daetwyler *et al.* (2013). This process of splitting, training, and predicting is typically repeated several times on the same dataset to get a combined or averaged measure of accuracy across different random partitions of the data.

Estimates of model accuracy by cross-validation are not perfect (Friedman *et al.* 2001). They are subject to sampling error as are any other statistic. They are also typically downwardly biased due to the fact that smaller training datasets are used for the cross-validation than in the actually application of a model, although this is minimized with Leave-One-Out Cross-validation. However in typical cases, this downward bias is the same for competing models and thus does not impact model choice.

More serious, though are potential biases in cross-validation estimates of model accuracy due to various forms of non-independence between the training and testing datasets. Several potential mistakes in cross-validation experiments are well known.

- **Biased testing data selection**. The individuals in the model testing partitions should have the same distribution of genetic (and environmental) relatednesses to the training population as individuals in the remaining target population (Amer and Banos 2010; Daetwyler *et al.* 2013). For example, if siblings or clones are present in the data, they should not be split between testing and training partitions unless siblings or clones of individuals in the training partition are also at the same frequency in the target population. Similarly, if the goal is to predict into a diverse breeding population, the cross-validation cannot be performed only on one F2 mapping population.
- **Non-independence of the testing and training datasets**. The observations used as testing data should be kept independent of the training data at all stages of the cross-validation procedure. For example, if data from individuals in the testing dataset are used to calculate estimated genetic values (EBVs) for model training, then the testing and training datasets are not independent, even if the testing individuals themselves are excluded from model training (Amer and Banos 2010). Similarly, if the predictands (e.g. EBVs of the testing partition) are estimated using data from the training partition, the testing and training partition are also not independent Legarra and Reverter (2018).
- **Pre-selection of features (e.g. markers) based on the full dataset before cross-validation**. All aspects of model specification and training that rely on the observed phenotypes should be performed only on the training partitions, without respect to the testing partition. For example, if a large number of candidate markers are available but only a portion will be included in the final model, the selection of markers (i.e. features) should be done using only the training partition of phenotypes and the selection itself should be repeated each replicate of the cross-validation on each new training dataset. If the feature selection is only done once on the whole dataset before cross-validation begins, this can lead to biased estimates of model accuracy (Friedman *et al.* 2001).

If these mistakes are avoided, cross-validation generally works well for comparing among single-trait methods, and in some cases for multi-trait methods. However, our goal in this paper is to highlight a challenge with using cross-validation to choose between single-trait and certain multi-trait method. In particular, when the goal of the multi-trait method is to leverage information from "secondary" traits measured on the target individuals to inform the prediction of their focal trait(s), standard cross-validation approaches fail. In this case, complete independence of the training and testing datasets with respect to all non-genetic factors is not possible. The specific information that we aim to use is only available on the individuals that we aim to predict into.

In the following sections, we first describe the opportunity offered by multi-trait genomic prediction models in this setting, and the challenge in evaluating them. We then develop a simulation strategy that highlights the extent of the problem. Finally, we propose two partial solutions that lead to fairly consistent model selections between single and multi-trait models under certain situations. Finally, we draw conclusions on when this issue is likely to arise and when it can be safely ignored.

GENERAL SETTING

Multi-trait genomic prediction is useful in two general settings: 1) When the overall value of an individual depends on each trait simultaneously (ex. fruit number and fruit size) and these traits are correlated, and 2) When a focal trait is difficult or expensive to measure on every individual, but other correlated traits are more readily available Thompson and Meyer (1986); Pszczola *et al.* (2013); Lado *et al.* (2018). While multi-trait models are clearly necessary in the first setting, in the second the value of the secondary traits depends on several factors including i) the repeatability of the focal and secondary traits, ii) the correlations among the traits and the *cause* of the correlations (i.e. genetic vs non-genetic), and iii) the relative expenses of collecting data on each trait.

Here we focus on the goal of predicting a single focal trait using information from both genetic markers (or pedigrees) and phenotypic information on other traits. Even within this context, there are also two distinct prediction settings: 1) Predicting the focal trait value for new individuals that are yet to be phenotyped for any of the traits, and 2) Predicting the focal trait value for individuals that have been partially phenotyped; phenotypic values for the secondary traits are known and we wish to predict the individual's genetic value for the focal trait. These settings were described by Burgueño *et al.* (2012) as CV1 and CV2, respectively, although those authors focused on multi-environment trials rather than single experiments with multiple traits per individual. The same naming scheme has since been extended to the more general multiple-trait prediction scenarios Lado *et al.* (2018).

The key difference between CV1 and CV2-style multi-trait prediction is that in the former, the secondary traits help refine estimates of the genetic values of *relatives* of the individuals we which to predict, while in the latter, the secondary traits provide information directly about the genetics of the focal individuals themselves. This direct information on the focal individuals is generally useful (as we demonstrate below). However, it comes with a cost for the *evaluation* of prediction accuracy by cross-validation. Since we do not know the true genetic values for the testing individuals, we must either use a model to estimate them or simply use their phenotypic value as a proxy. Unfortunately, if we use our genetic model to estimate these values, we are breaking the independence between the testing and training data, and therefore have biased estimates of cross-validation accuracy. On the other hand, if we simply use the phenotypic values of the focal trait as our predictand, these may be biased towards or away from the true genetic values depending on the non-genetic correlation between the focal and secondary traits because these traits are measured on the same individuals. This leads to either over- *or under*-estimation of the prediction accuracy of our multi-trait models. In realistic scenarios, this can lead users to select worse models.

MATERIALS AND METHODS

We used a simulation study to explore conditions when naive cross-validation experiments as described above lead to sub-optimal choices between single and multi-trait genomic prediction methods. Our simulations were designed to mimic the process of using cross-validation to compare single and multi-trait models based on their prediction accuracies. We repeated this simulation across scenarios with different genetic architectures for two traits: a single "focal" trait and a single "secondary" trait. Specifically, we modified the genetic determinacy and correlation structure of the two traits. These are the most important parameters for determining the relative efficiencies of single- and multi-trait prediction models (Thompson and Meyer 1986). Sample size and level of genomic relatedness will also affect the comparisons, but are likely to change the relative performances of the models and the accuracy of cross-validation quantitatively but not qualitatively.

We created a population of $n = 1000$ individuals with known genomic relationships, specified by the genomic covariance matrix $\mathbf{K}$. We then simulated two traits for each individual from a multivariate normal distribution with defined percentages of genetic and non-genetic variation ($H^2 = \{0.2, 0.6\}$), and defined genetic and non-genetic correlations between the traits $\rho_g = \{0, 0.3, 0.6\}$, $\rho_R = \{-0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6\}$. In particular, we set:

$$\begin{aligned} &\mathbf{Y} = \mathbf{U} + \mathbf{E}, \qquad \mathbf{U} \sim \mathrm{MN}(\mathbf{0}, \mathbf{K}, \mathbf{G}), \qquad \mathbf{E} \sim \mathrm{MN}(\mathbf{0}, \mathbf{I}_n, \mathbf{R}) \\ &\mathbf{G} = \begin{bmatrix} \mathbf{g}_{11} & \mathbf{g}_{12} \\ \mathbf{g}_{21} & \mathbf{g}_{22} \end{bmatrix} = \begin{bmatrix} H_1^2 & \rho_g H_1 H_2 \\ \mathbf{g}_{12} & H_2^2 \end{bmatrix} \\ &\mathbf{R} = \begin{bmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} \\ \mathbf{r}_{21} & \mathbf{r}_{22} \end{bmatrix} = \begin{bmatrix} (1-H_1^2) & \rho_R\sqrt{(1-H_1^2)(1-H_2^2)} \\ \mathbf{r}_{12} & (1-H_2^2) \end{bmatrix} \end{aligned} \tag{1}$$

where MVN(.) is the Matrix normal distribution, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2]$ are the phenotypic values for the two traits in the n individuals, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2]$ are the true genetic values for the two traits, and $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2]$ are the true non-genetic deviations for the two traits. We repeated this process 500 times for each of the 42 combinations of the genetic architecture parameters. To improve the consistency of the simulations, we used the same draws from a standard-normal distribution for all 42 parameter combinations, but new draws for each of the 500 simulations.

For computational convenience, we used as a genomic relationship matrix the pedigree-based additive relationship matrix for a half-sib design with 10 individuals per half-sib family (and therefore 100 independent families), so that the diagonal of $\mathbf{K}$ was always 1 and off-diagonal elements were either 0.25 or 0. While this matrix is not typical of a true "genomic" or realized genomic kernel matrix in real data, its simple structure allowed the numerical calculations to be performed with sparse matrices, greatly increasing the number of simulations we could run. Most common single and multi-trait genomic prediction methods such as GBLUP, the Bayes Alphabet methods, or RKHS methods can be written as a form of model 1 if the true genetic architecture is known. In our case, since we simulated based on our $\mathbf{K}$, it is the "true" genomic relatedness matrix for our data. We also explored other genetic architectures (increasing the within-family relatedness to 0.5 or decreasing it to 0.1) with no qualitative changes in the results.

After creating the 1000 simulated individuals, we divided them 90:10 into a training partition of 900 individuals, and a testing partition of 100 individuals. For simplicity, we chose one individual from each of the 100 families to use as the testing partition. We arranged the rows of $\mathbf{Y}$ so that the testing individuals were first, and correspondingly partitioned $\mathbf{K}$ into:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{nn} & \mathbf{K}_{no} \\ \mathbf{K}_{on} & \mathbf{K}_{oo} \end{bmatrix}. \tag{2}$$

Here and below, the subsctipt $_n$ refers to the testing partition (i.e. "new" individuals) and the subscript $_o$ refers to the training partition (i.e. "old" individuals). We use the hat symbol (ˆ) to denote parameter estimates or predictions.

We then fit single- and multi-trait linear mixed models to the training data and used these model fits to predict the genetic values for the focal trait (trait 1) in the testing partition.

Specifically, for the single-trait method we fit a univariate linear mixed model to the training data $\mathbf{y}_{o1}$:

$$\mathbf{y}_{o1} = \mu_1 + \mathbf{u}_{o1} + \mathbf{e}_{o1}, \quad \mathbf{u}_{o1} \sim \mathrm{N}(0, \mathbf{g}_{11}\mathbf{K}_{oo}), \quad \mathbf{e}_{o1} \sim \mathrm{N}(0, \mathbf{r}_{11}\mathbf{I}_{n_o}) \tag{3}$$

and extracted the BLUPs $\hat{\mathbf{u}}_{o1}$. Note: an expanded version of these derivations are provided in the Appendix. We used a slightly modified version of the `relmatLmer` function of the `lme4qtl` R package (Bates *et al.* 2015; Ziyatdinov *et al.* 2018) to fit the model by Restricted Maximum Likelihood (REML). We then calculated predicted genetic values for the testing partition $\mathbf{u}_{n1}$ as:

$$\hat{\mathbf{u}}_{n1}^{(1)} | \hat{\mathbf{u}}_{o1} = \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\hat{\mathbf{u}}_{o1}. \tag{4}$$

For the multi-trait model, we stacked the vectors of the two traits in the training dataset into the vector $\mathbf{y}_o = \begin{bmatrix} \mathbf{y}_{n1} \\ \mathbf{y}_{n2} \end{bmatrix}$ and fit:

$$\mathbf{y}_o = \boldsymbol{\mu} + \mathbf{u}_o + \mathbf{e}_o, \quad \mathbf{u}_o \sim \mathrm{N}(\mathbf{0}, \mathbf{G} \otimes \mathbf{K}_{oo}), \quad \mathbf{e}_o \sim \mathrm{N}(\mathbf{0}, \mathbf{R} \otimes \mathbf{I}_{n_o}) \tag{5}$$

using the `relmatLmer` function, extracted estimates $\hat{\mu} = [\hat{\mu}_1^\intercal, \hat{\mu}_2^\intercal]^\intercal$, $\hat{\mathbf{G}}$, $\hat{\mathbf{R}}$, and BLUPs $\hat{\mathbf{u}}_o$.

To make predictions of the genetic values for the focal trait in the validation partition in the CV1 case without use of $\mathbf{y}_{n2}$, we calculated:

$$\hat{\mathbf{u}}_{n1}^{(2)} | \hat{\mathbf{u}}_{o1} = \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\hat{\mathbf{u}}_{o1} \tag{6}$$

which has the same form as for the single trait model, but the input BLUPs $\hat{\mathbf{u}}_{o1}$ are different.

To make predictions of the genetic values for the focal trait in the testing partition in the CV2 case with the phenotypic observations of the secondary trait $\mathbf{y}_{n2}$, we used a two step method. First, we estimated $\hat{\mathbf{u}}_o$ above based on both traits in the training data. Then we combined these estimates with the observed phenotypes of the testing data to calculate genetic predictions for the testing data:

$$\begin{aligned}\hat{\mathbf{u}}_{n1}^{(3)}|\mathbf{y}_{n2}, \hat{\mathbf{u}}_o &= \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\hat{\mathbf{u}}_{o1} \\ &\quad + \hat{\mathbf{g}}_{12}(\mathbf{K}^{-1})_{nn}(\widehat{\mathbf{V}}_c)^{-1}(\mathbf{y}_{n2} - \hat{\mu}_2 - \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\hat{\mathbf{u}}_{o2}), \quad (7) \\ \widehat{\mathbf{V}}_c &= \hat{\mathbf{g}}_{22}(\mathbf{K}^{-1})_{nn} + \hat{\mathbf{r}}_{22}\mathbf{I}_n.\end{aligned}$$

This two-step method will be slightly less efficient than a one-step method that used $\mathbf{y}_{n2}$ during the estimation of $\hat{\mathbf{u}}_o$, but the difference in computational efficiency was minimal in our tests, and this two step method gives same genetic predictions and is much easier to implement in breeding programs because no genotype or phenotype data of the evaluation individuals is needed during the model training stage.

We measured the accuracy of these three predictions by calculating the correlation between the prediction $\hat{\mathbf{u}}_{n1}^{(i)}$ and four predictands:

- $\mathbf{u}_{n1}$: The true genetic value over the 500 simulations.
- $\mathbf{y}_{n1}$: The phenotypic values of the testing individuals.
- $\tilde{\mathbf{u}}_{n1}$: The estimated genetic values of the validation individuals using the full dataset (including $\mathbf{y}_{n1}$).
- $\mathbf{y}_{x1}^k$: The phenotypic values of new individuals from the target population that are close relatives of the validation individuals but experienced different micro-environments. $k = \{0.25, 0.5, 1\}$ denotes the level of relatedness between these individuals and the corresponding individuals in the validation partition. These new individuals were simulated in parallel with the original n individuals, but were not used in either the model training or genetic value prediction calculations.

For the two accuracy measures that use phenotypic values as predictands, we "corrected" the correlations by dividing by $\sqrt{\hat{h}^2}$ to account for the larger variance of $\mathbf{y}_{n1}$ relative to $\mathbf{u}_{n1}$, which impacts the denominator of the correlation (Daetwyler *et al.* 2013). In cases when $\hat{h}_1^2 < 0.02$, we set $\hat{h}_1^2 = 0.2$ for this correction.

For each genetic architecture setting, we declared the "best" prediction method to be the one with the highest average correlation with the true genetic values across the 500 simulations. Then we counted the proportion of the simulations in which this "best" method actually had the highest estimated accuracy when scored against $\mathbf{y}_{n1}$, $\tilde{\mathbf{u}}_{n1}$, or $\mathbf{y}_{x1}^k$.

Scripts for running all simulations and analyses described here are available at https://github.com/deruncie/multiTrait_crossValidation_scripts.

RESULTS

Although we ran simulations for two levels of genetic determinacy for the focal trait ($H_1^2 = \{0.2, 0.6\}$) we present results only for $H_1^2 = 0.2$. This is the "most-difficult" setting for prediction–when the genetic determinacy of the trait is low–but also the setting when we would expect the greatest benefit of using multi-trait models. However, results for $H_1^2 = 0.6$ gave qualitatively similar results, just with higher overall prediction accuracies of all methods.

Accuracy of single and multi-trait methods in simulated data

With $H_1^2 = 0.2$ the true accuracy of prediction $cor(\hat{\mathbf{u}}_{n1}, \mathbf{u}_{n1})$ was low (~ 0.25) for all methods (Figure 1). Prediction accuracy for the single-trait method was constant across settings with different correlation structures because information from the secondary trait was not used.

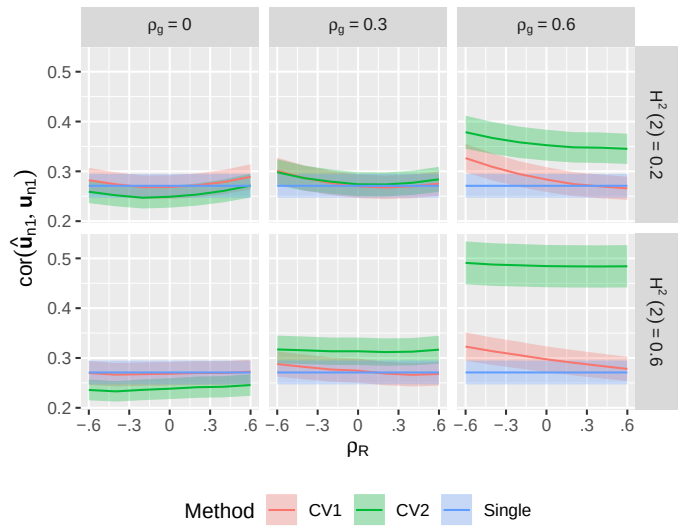

Figure 1 Actual prediction accuracy of single-trait and multi-trait prediction methods in simulated data. 500 simulations were run for each genetic determinacy of the secondary trait ($H_2^2 = \{0.2, 0.6\}$), and each combination of genetic and non-genetic correlation between the two traits ($\rho_g = \{0, 0.3, 0.6\}, \rho_R = \{-0.6, -0.4, -0.2, 0, 0.2, 0, 4, 0.6\}$), all with $H_1^2 = 0.2$. For each simulation, we used the 900 training individuals to fit linear mixed models (either single or multi-trait), predicted the genetic values of the 100 testing individuals, and then measured the Pearson's correlation between the predicted ($\hat{\mathbf{u}}_{n1}$) and true ($\mathbf{u}_{n1}$) genetic values. In the CV1 method, we used only information on the testing individuals to calculate $\hat{\mathbf{u}}_{n1}$. In the CV2 method, we used the training individuals to calculate $\hat{\mathbf{u}}_o$ and combined this with the observed phenotypes for the secondary trait on the testing individuals ($\mathbf{y}_{n2}$). Curves show the average correlation for each method across the 500 simulations. Ribbons show $\pm 1.96 \times SE$ over the 500 simulations.

The "standard" muti-trait model (i.e. CV1-style) that used phenotypic information only on the training partition slightly outperformed the single-trait model in some settings, more-so when the genetic and non-genetic correlations between traits were large and opposing and when the genetic determinacy of the secondary trait was high (Thompson and Meyer 1986). However it performed worse whenever the genetic correlation between traits was low. This was caused by inaccuracy in the estimation of the two covariance parameters ($\hat{\mathbf{g}}_{12}, \hat{\mathbf{r}}_{12}$). Neither multi-trait model performed worse than the single-trait model when the true $\mathbf{G}$ and $\mathbf{R}$ matrices were used Supplemental Figure 1. But in real data, mult-trait models require estimating more (co)variance parameters and therefore can show reduced performance when data are limited.

The CV2-style multi-trait method which leverages additional phenotypic information on the secondary trait from the testing partition itself showed dramatic improvements in prediction accuracy whenever genetic correlations among traits were large, irregardless of the non-genetic correlation between the traits. This is similar

to the benefits seen by Rutkoski *et al.* (2016) and Lado *et al.* (2018). When the genetic determinacy of the secondary trait was high, the improvement in prediction accuracy was particularly dramatic (increasing to ~ 0.5). This is the potential advantage of incorporating secondary traits into prediction methods. However, the CV2 method also requires estimating $\mathbf{G}$ and $\mathbf{R}$, and its performance was lower than the single-trait method whenever genetic correlations were low.

Therefore, multi-trait methods will not always be useful and it is important to test the relative performance of the different methods in real breeding scenarios. Unfortunately, we never know the true genetic values ($\mathbf{u}_{n1}$), and so must use proxy predictands to evaluate our methods in real data (Daetwyler *et al.* 2013; Legarra and Reverter 2018). In Figures 2A-B, we compare the prediction accuracies of the three methods using two candidate predictands: the observed phenotypic values ($\mathbf{y}_{n1}$) and estimated genetic values from a joint model fit to the complete dataset ($\tilde{\mathbf{u}}_{n1}$).

Using the observed phenotypic values ($\mathbf{y}_{n1}$) as the predictand, the estimated accuracy of both the single-trait and CV1-style multi-trait prediction methods consistently under-estimated their true prediction accuracies. This is expected because in this setting 80% of the phenotypic variation is non-genetic and cannot be predicted based on relatives alone. We therefore follow common practice to report a "corrected" estimate of the prediction accuracy by dividing by $\sqrt{\hat{h}^2}$ in Figure 2A. Since the estimate of h^2 is approximately the same for the three prediction methods, this correction factor does not affect the model selection between the methods.

In contrast, the estimated accuracy of the CV2-style multi-trait method varied dramatically across simulated datasets. We tended to overestimate the true accuracy when both genetic and non-genetic correlations were large and in the same direction, and dramatically underestimate the true accuracy when the two correlations were opposing. Importantly, there are situations where the CV2-style method appears to perform worse than the single-trait method based on $\mathbf{y}_{n1}$ but actually performs better, and situations where the CV2-style method appears to perform better and actually performs worse (compare solid to dashed green lines). Therefore, the observed phenotypic values are not reliable predictands to evaluate CV2-style methods when the intent is to estimate true genetic values.

On the other hand, using estimated genetic values from a joint model fit to the complete dataset ($\tilde{\mathbf{u}}_{n1}$) as the predictand led to dramatic over-estimation of the true prediction accuracy for all methods. This is also expected because the training data are used both to train the prediction model *and also* to create the testing dataset, a clear violation of the cross-validation rules that these datasets must be kept separate at all stages of the analysis. Again, the bias was most severe for the CV2-style method. Since this method is clearly invalid, we do not consider it further.

Effects of predictand on model selection

To demonstrate the impact of biased estimates of model accuracy using $\mathbf{y}_{n1}$ on the effectiveness of model selection, we assessed in each simulation whether the single-trait or multi-trait methods had a higher *estimated* accuracy, and compared this result to the *true* prediction accuracy of the method for that simulation.

Figure 3A shows that selecting between the single-trait and CV1-style multi-trait based on estimated accuracy using $\mathbf{y}_{n1}$ generally works well. Whenever one method is clearly better, we are able to choose that method $> 50\%$ of the time. But we never choose correctly $< 50\%$ of the time, even when the methods are approximately equivalent.

In contrast, when selecting between the single-trait and CV2-style multi-trait methods based on estimated accuracy using $\mathbf{y}_{n1}$, the differential bias in estimated accuracy between the two methods frequently lead to sub-optimal model selectiosn (Figure 3B). With opposing genetic and non-genetic covariances between the two traits, the better model was only chosen $\sim 10 - 15\%$ of the time. In these situations, using $\mathbf{y}_{n1}$ to select a prediction method will obscure real opportunities to enhance prediction accuracy using multi-trait prediction models.

Alternative estimates of multi-trait prediction accuracy

The CV2-style prediction method can be powerful because $\mathbf{y}_{n2}$ provides information on the genetic value of the testing individuals themselves (through $\mathbf{u}_{n2}$), while $\mathbf{y}_{o1}$ only provides indirect information on the genetic values of the testing individuals through the relatives. However, estimating prediction accuracy using $\mathbf{y}_{n1}$ fails for the CV2-style prediction method because both the focal and secondary traits are observed on the same individual and therefore share the same non-genetic sources of variation. Since the CV2 method uses $\mathbf{y}_{n2}$, non-genetic deviations for the secondary trait $\mathbf{e}_{n2}$ push $\hat{\mathbf{u}}_{n1}$ either towards or away from $\mathbf{y}_{n1}$ depending on the estimated correlation $\hat{\mathbf{r}}_{12}$. This either inflates or deflates the estimated accuracy, leading to incorrect model choices.

Corrected accuracy estimates. In principle, we can correct for this bias by estimating its magnitude, and calculating an adjustment factor for the estimated accuracy ($cor(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{y}_{n1})$) to better approximate the true accuracy ($cor(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{u}_{n1})$). This is similar to the semi-parametric accuracy estimates presented by Legarra and Reverter (2018), and the "correction" of accuracy estimates by $1/\sqrt{\hat{h}^2}$ used above to account for the difference in variance between $\mathbf{y}_{n1}$ and $\mathbf{u}_{n1}$. As we derive in the Appendix, the difference between the true correlation from a CV2-style methods and its CV2 cross-validation estimate when a single secondary trait is used is:

$$\frac{\hat{\mathbf{g}}_{12}\mathbf{r}_{21}}{\sqrt{var(\hat{\mathbf{u}}_{n1}^{(3)})var(\mathbf{y}_{n1})}}\frac{tr(\mathbf{S}(\mathbf{K}^{-1})_{nn}\widehat{\mathbf{V}}_c^{-1}\mathbf{K}_{nn})}{n-1}. \tag{8}$$

with $\mathbf{V}_c$ defined above and $\mathbf{S} = \mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\intercal}{n}$. This bias is dominated by the product $\hat{\mathbf{g}}_{12}\mathbf{r}_{21}$ (as the second term does not involve these parameters), which is large and positive (i.e. true accuracy is over-estimated) when $\hat{\mathbf{g}}_{12}$ and $\mathbf{r}_{12}$ are large and in the same direction, and large and negative (i.e. true accuracy is underestimated) when these covariances are in opposite directions. Given this result, we can correct $cor(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{y}_{n1})$ by subtracting 8 from the estimated correlation, again corrected by $1/\sqrt{\hat{h}^2}$ (Figure 4).

Clearly, the quality of this correction will depend on the accuracy of $\hat{\mathbf{g}}_{12}$ and $\hat{\mathbf{r}}_{12}$ as estimates of $\mathbf{g}_{12}$ and $\mathbf{r}_{12}$. In Figure 4, we show that the corrected correlation estimate has greatly reduced bias, particularly the dependence of the bias on the non-genetic covariance between the traits $\mathbf{r}_{12}$. However the correction is not perfect. Corrected accuracy estimates still tend to overestimate the true accuracy, particularly when the genetic covariance is small. This over-estimation is caused by error in $\hat{\mathbf{G}}$ and $\hat{\mathbf{R}}$ as estimates of the true covariances. The correction factor is nearly perfect when the true covariance matrices are used in place of their estimates Supplemental Figure 2. Similarly, if we increase the size of our simulated populations to 5000 individuals, which improves the quality of the covariance estimates, the correction factor is much more accurate Supplemental Figure 3.

Using the corrected accuracy estimates, we are more successful at selecting the best model over the range of genetic architectures

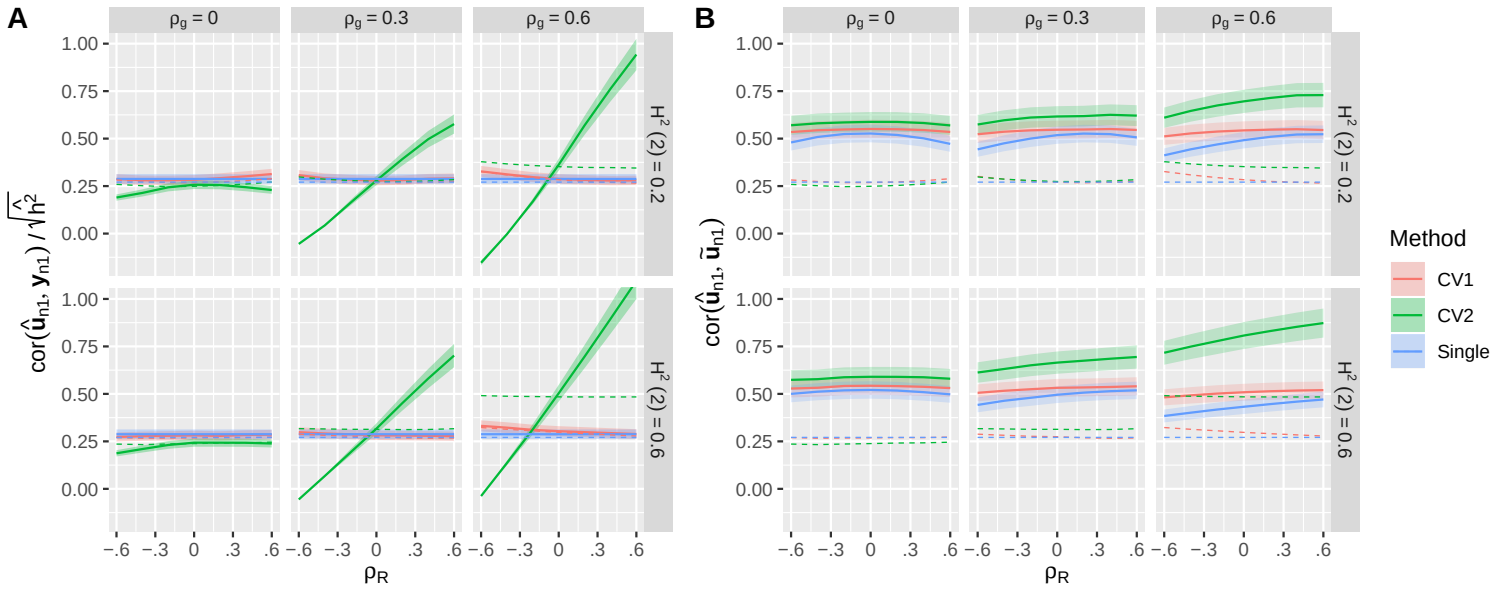

Figure 2 Estimated prediction accuracies based on candidate predictands. For the same set of simulations described in Figure 1, we estimated the prediction accuracies of the three methods using two different candidate predictands: (**A**) The observed phenotypic value $\mathbf{y}_{n1}$ for each training individual with the correlation corrected by $1/\sqrt{\hat{h}^2}$, or (**B**) An estimate of the genetic value of each training individual based on the BLUPs calculated using the complete phenotype data ($\tilde{\mathbf{u}}_{n1}$). Solid lines in each panel show the average *estimated* accuracy for each method across the 500 simulations. Ribbons show $\pm 1.96 \times SE$ over the 500 simulations. Dotted lines show the average *true* accuracy from Figure 1.

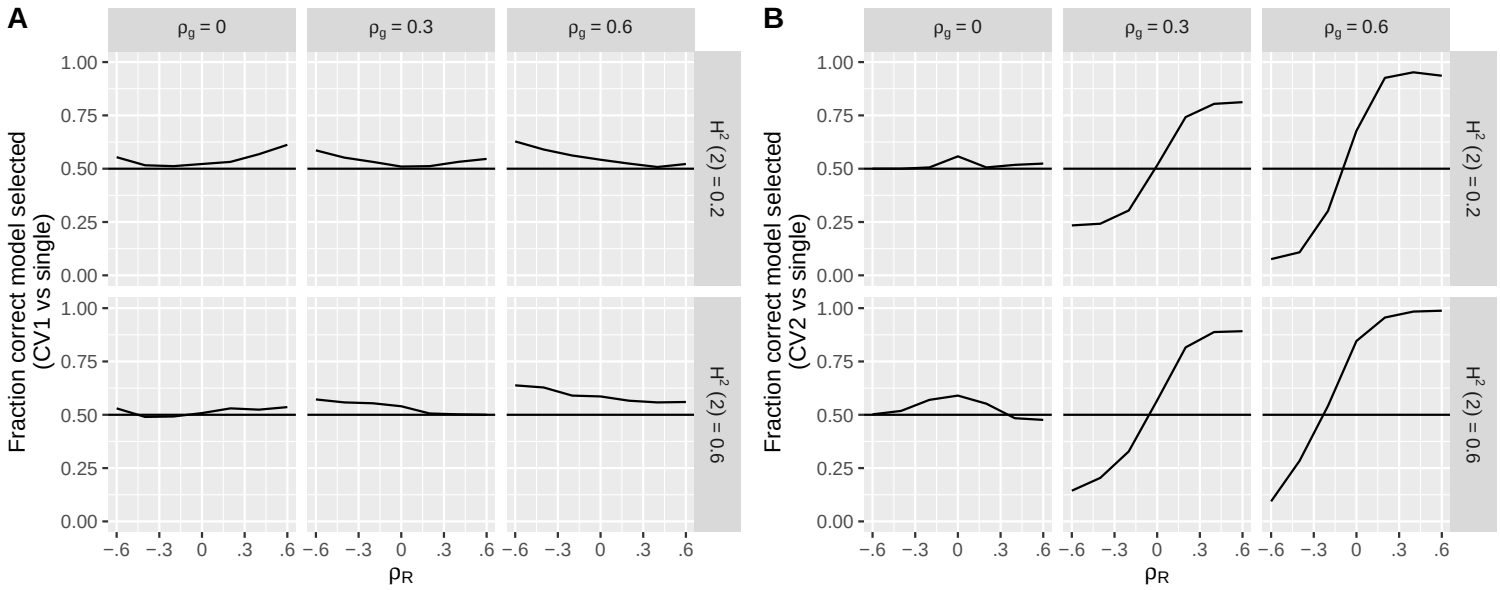

Figure 3 Impact of using phenotypic data to select between single-trait and multi-trait prediction methods. For each of the 500 simulations per genetic architecture described in Figure 1, we compared the estimated accuracy of a multi-trait prediction to the single-trait prediction. We then calculated the fraction of times that the selected model had higher average true accuracy in that setting (as shown in Figure 1). (**A**) CV1-style multi-trait method vs single-trait prediction. (**B**) CV2-style multi-trait method vs single-trait prediction.

(Figure 4B). The frequency of selecting the correct model never drops below 50%. However we only see a better than random selection when the true genetic covariance between the traits is non-zero.

CV2* cross-validation strategy. Since the biased estimate of prediction accuracy for CV2-style methods is due to non-genetic correlations between $\mathbf{y}_{n2}$ used for prediction and the predictand $\mathbf{y}_{n1}$, an alternative strategy, which we call CV2*, is to use phenotypic information on close relatives of the testing individuals ($\mathbf{y}_{x1}$) to validate the model predictions in place of $\mathbf{y}_{n1}$. These "surrogate" validation individuals must also be excluded from the model training and raised so that they do not share the same non-genetic deviations as the testing individuals: $cor(\mathbf{e}_{x1}, \mathbf{e}_{n1}) = 0$. Therefore, $\hat{\mathbf{u}}_{n1}$ will not be artificially pushed towards or away from $\mathbf{u}_{x1}$ (measured on relatives) by $\mathbf{y}_{n2}$ (measured on testing individuals), preventing this source of bias in the estimated accuracy.

If genetic twins (i.e. clones) of the testing individuals are available as the surrogate validation individuals, then their genetic values are identical (by definition) to those of the testing individuals ($\mathbf{u}_{x1} = \mathbf{u}_{n1}$). Thus, their phenotypic values $\mathbf{y}_{x1}$ are as good a predictand as $\mathbf{y}_{n1}$ for evaluating the accuracy of the CV2 method. If only more distant relatives are available, (e.g. $max(\mathbf{K}_{nx}) = 0.5$ or 0.25) the estimate of the CV2 method's accuracy declines (Figure 5A). But it is still generally higher than the single-trait method whenever the CV2 method is actually better (as defined by Figure 1). Recall that in this simulation, families were composed of half-sibs. Therefore when $max(\mathbf{K}_{nx}) = 0.25$ the surrogate testing individuals were no more closely related to the actual testing individuals than members of the training set

Importantly, using surrogate validation individuals rather than the testing individuals themselves provides a consistent method for selecting between the single-trait and CV2-style multi-trait prediction method, with the optimal selection probability never dropping below 50% (Figure 5B). The power to select the better model is highest when clones are available, but is still elevated above 50% for full-sibs and in some settings for half-sibs. Therefore, CV2* is valid as a method for evaluating when CV2-style multi-trait prediction is useful.

DISCUSSION

Our study highlights a potential pitfall in using cross-validation to estimate the accuracy of multi-trait genomic prediction methods. When secondary traits are used to aid in the prediction of focal traits, and these secondary traits are measured on the individuals to be tested, cross-validation evaluated against phenotypic observations can be severely biased and result in poor model choices. Unfortunately, we never know the true genetic value of any individual, and therefore can only evaluate our models with phenotypic data (since multi-trait-derived estimated genetic values are even more severely biased as we demonstrated above (Figure 2B). We cannot find earlier discussions of this problem in the literature. However a growing number of studies do aim to use cheap or early-life traits to improve predictions of genetic worth for individuals in later-life traits (ex. Pszczola *et al.* 2013; Rutkoski *et al.* 2016; Fernandes *et al.* 2017; Lado *et al.* 2018). Therefore the issue is becoming more important.

The problematic bias in the cross-validation-based accuracy estimates is caused by non-genetic correlations between the predictors that we want to use (i.e. the secondary traits) and our best predictand (the phenotypic value of the trait in the testing individuals) – non-genetic correlations between two traits measured on the same individual are expected. However, in some cases this correlation is zero by construction, and standard cross-validation approaches can be valid. For example, in the original description of the CV2 cross-validation method by Burgueño *et al.* (2012), each trait was measured in a different environment. In this case, the traits were measured on different plants and therefore did not share any non-genetic correlation. Also, CV1-style methods do not suffer from this problem because phenotypic information on the secondary traits *in the testing individuals* is not used for prediction. Similarly, this bias does not occur when the target of prediction is the phenotypic value itself (rather than the individual's genetic value). For example, in medical genetics the aim is to predict whether or not a person will get a disease or not, not her genetic propensity to get a disease had she been raised in a different environment (ex Spiliopoulou *et al.* 2015; Dahl *et al.* 2016).

We note that the common strategy of two-step genome selection: using single-trait methods to calculate estimated genetic values for each line:trait and then using these estimated genetic values as training (and validation) data, does not get around the problem identified here. Using estimated genetic values instead of phenotypic values will tend to increase the genetic repeatability of the training and validation values, and therefore increase the overall prediction accuracy of all methods. But these estimated genetic values will still be biased by the non-genetic variation, and the biases across traits will still be correlated by the non-genetic correlations. Therefore the same issue will arise.

Also, while we have used a GBLUP-like genomic prediction method for the analyses presented here, the same result will hold for any multi-trait prediction method that aims to use information from $\mathbf{y}_{n2}$ when there are non-genetic correlations with $\mathbf{y}_{n1}$, i.e. any method that is evaluated with the CV2 cross-validation method (Calus and Veerkamp 2011; Jia and Jannink 2012; Fernandes *et al.* 2017). This includes multi-trait versions of the Bayes Alphabet methods (Calus and Veerkamp 2011), or neural network or Deep Learning methods (Montesinos-López *et al.* 2018).

We proposed two partial solutions to this problem. The first partial solution is to estimate a correction factor for the CV2-style multi-trait prediction accuracy estimate, and used this semi-parametric corrected estimate to compare and validate models. This is similar to the approach of Legarra and Reverter (2018) for single-trait models. We show that this correction factor can work, particularly if the covariances among traits are well estimated. We only derive this correction method for prediction methods based on linear mixed effect models with a single known genetic covariance structure (i.e. GBLUP and RKHS-style methods), although the approximation $\frac{\hat{\mathbf{g}}_{12}\hat{\mathbf{r}}_{12}}{\sqrt{(var(\hat{\mathbf{u}})var(\mathbf{u})}}$ will probably be approximately correct for other methods. However, when the covariances are poorly estimated, the correction factor can still lead to biased estimates of model accuracy. We are currently investigating whether Bayesian methods that sample over this uncertainty can be useful in which we calculate the correction factor each iteration, and will implement this method in JWAS (Cheng *et al.* 2018). Also, this method is semi-parametric, so relies on assumptions about the data (i.e. multivariate normality) that may not be accurate, and so loses some of the guarantees of reliability that completely non-parametric cross-validation methods can claim.

As an alternative, we proposed the CV2* cross-validation method, a fully non-parametric approach for assessing CV2-style multi-trait prediction accuracy. CV2* involves using phenotypic values of the focal trait from relatives of the testing individuals in place of the phenotypic values of that trait from the testing individuals themselves. If the close relatives are raised independently,

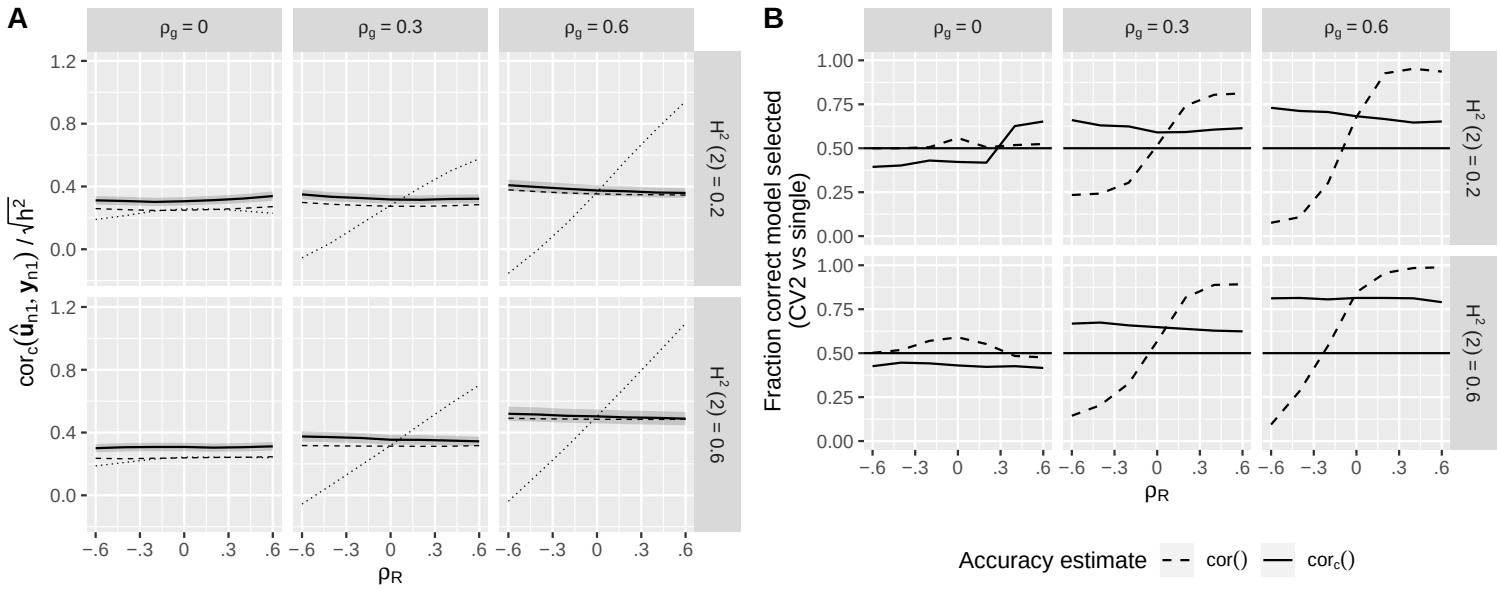

Figure 4 Estimated prediction accuracies and model selection accuracies for CV2-style methods after semi-parametric correction. (**A**) Estimated prediction accuracy. Ribbons show $\pm 1.96 \times SE$ over the 500 simulations. (**B**) Percentage of the simulations in which the CV2-style methods estimated accuracy was greater than the single-trait estimated accuracy, and the CV2-style method was actually more accurate. Solid curves: corrected estimates of prediction accuracy. Dashed curves: true prediction accuracy. Dotted curves: uncorrected estimates of prediction accuracy based on $\mathbf{y}_{n1}$ (mirroring Figure 3).

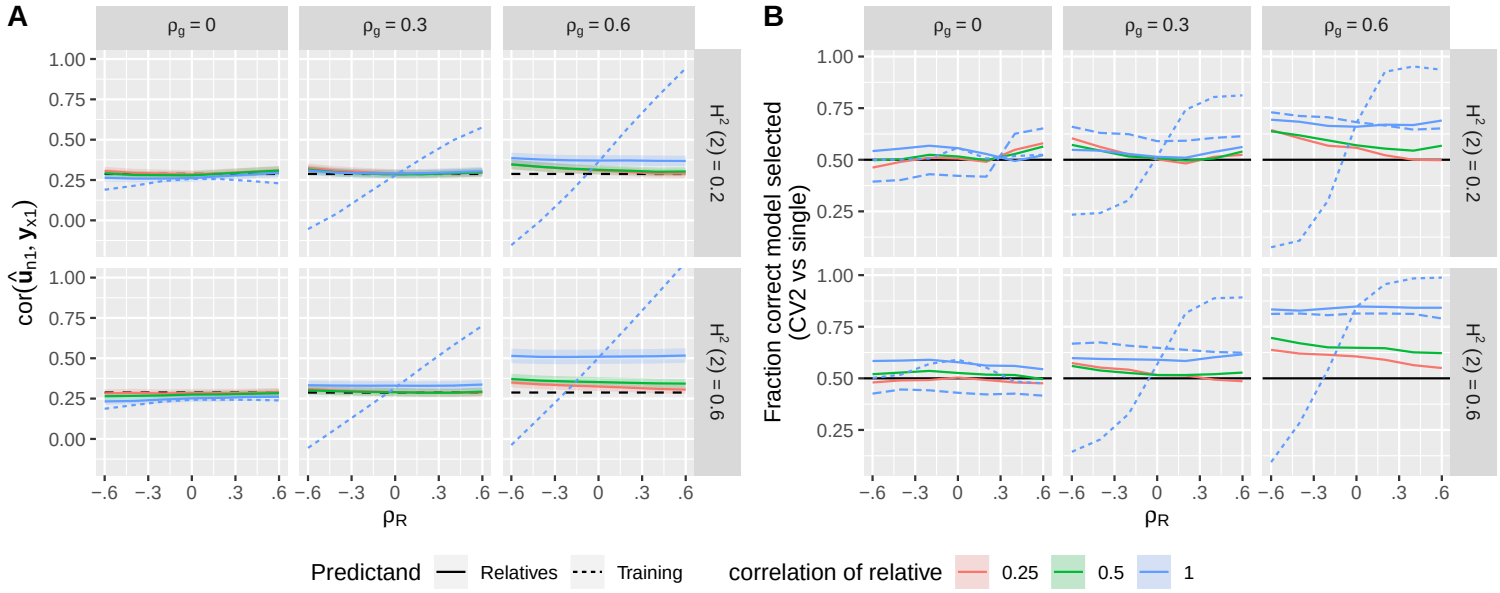

Figure 5 Estimate prediction accuracies and model selection accuracies for CV2-style methods evaluated on the phenotypic values of close relatives. (**A**) Solid curves: the estimated prediction accuracies of the CV2 method evaluated against the phenotypic values for relatives of the testing individuals ($\mathbf{y}_{x1}$). Different colors represent different classes or relatives; blue: clones, green: full-sibs, red: half-sibs. Dotted lines: estimated accuracies of the CV2 (blue) and single-trait (black) methods using the phenotypes of the testing individuals themselves ($\mathbf{y}_{n1}$). (**A**) Solid curves: the fraction of the 500 simulations in the better method (between CV2 and single-trait) for predicting the true genetic values was selected based on the phenotypes of relatives of the testing individuals. Dotted lines: the fraction of correct models selected based on the phenotypes of the testing individuals themselves (mirroring Figure 3). Ribbons show $\pm 1.96 \times SE$ over the 500 simulations

they will not share non-genetic variation, removing the source of bias in the cross-validation estimate (Figure 5A).

The CV2* method works best when clones of the testing individuals are available. With clones, secondary trait phenotypes of the testing individuals can be used directly to predict focal trait genetic values of their clones because the genetic values are identical. Replicates of inbred lines are frequently used in plant breeding trials (Bernardo 2002). In this case, all replicates should be held-out as a group from the training data. Then the replicates can be partitioned again into two sets; secondary trait phenotypes from one set can be incorporated into the genetic value predictions for the lines, and these predictions evaluated against the phenotypic values of the other set. To compare this estimate of CV2-style prediction accuracy to the prediction accuracy for a single-trait method, the single-trait method's predictions should be compared against the same set of replicates of each line (i.e., not a joint average over all replicates of the line as would be typical for single-trait cross-validation).

Clones are less common outside of plant breeding, so more distant relatives need to be used instead if CV2* is necessary. In this case, the estimated prediction accuracies of CV2-style methods will be downwardly biased. Additionally, CV2* can be difficult to implement for genomic prediction methods that do not rely on a pedigree because it is not necessarily clear *a priori* which individuals are close relatives. The closest relatives are those that share genotypes at the most important QTL, but this information is not known when splitting a dataset into training and testing portions. In this case, the best choice will likely be to choose pairs of individuals with the highest pairwise relatedness estimated uniformly from marker data (i.e. the realized genomic relatedness used as an imput to GBLUP methods). Once each individual is paired with its closest genomic relative, pairs can be assigned together to cross-validation folds. Then secondary trait phenotypes of one individual can be used to predict genetic values of the other. These predictions can then be scored against the second individual's focal trait phenotype to estimate the CV2 prediction accuracy and select an optimal prediction method. Alternatively, in these cases, the corrected estimate of accuracy from the standard CV2 cross-validation method can be used if the expected accuracy of the covariances is high.

Overall, we expect that multi-trait methods for genomic prediction carry great promise to accelerate both plant and animal breeding. However there is a need to design better methods to evaluate and train the prediction methods to ensure that models can be accurately compared.

ACKNOWLEDGMENTS

We would like to thank Erin Calfee and Graham Coop for suggesting the CV2* method.

SUPPLEMENTAL FIGURES

Supplemental Figure 1 Actual prediction accuracy of single-trait and multi-trait prediction methods in simulated data when G and R are known. 500 simulations were run for each genetic determinacy of the secondary trait ($H_2^2 = \{0.2, 0.6\}$), and each combination of genetic and non-genetic correlation between the two traits ($\rho_g = \{0, 0.3, 0.6\}, \rho_R = \{-0.6, -0.4, -0.2, 0, 0.2, 0, 4, 0.6\}$), all with $H_1^2 = 0.2$. For each simulation, we used the 900 training individuals to fit linear mixed models (either single or multi-trait) conditioning on the true values for **G** and **R**, predicted the genetic values of the 100 testing individuals, and then measured the Pearson's correlation between the predicted ($\hat{\mathbf{u}}_{n1}$) and true ($\mathbf{u}_{n1}$) genetic values. In the CV1 method, we used only information on the testing individuals to calculate $\hat{\mathbf{u}}_{n1}$. In the CV2 method, we used the training individuals to calculate $\hat{\mathbf{u}}_o$ and combined this with the observed phenotypes for the secondary trait on the testing individuals ($\mathbf{y}_{n2}$). Curves show the average correlation for each method across the 500 simulations. Ribbons show $\pm 1.96 \times SE$ over the 500 simulations.

Supplemental Figure 2 Estimated prediction accuracies and model selection accuracies for single-trait and multi-trait prediction methods after semi-parametric correction when when G and R are known. Ribbons show $\pm 1.96 \times SE$ over the 500 simulations.

Supplemental Figure 3 Estimated prediction accuracies and model selection accuracies for single-trait and multi-trait prediction methods in larger datasets after semi-parametric correction when when G and R are known. Simulations were run with 5000 individuals, with 500 held out as testing individuals. Ribbons show $\pm 1.96 \times SE$ over the 500 simulations.

LITERATURE CITED

Amer, P. R. and G. Banos, 2010 Implications of avoiding overlap between training and testing data sets when evaluating genomic predictions of genetic merit. Journal of Dairy Science **93**: 3320–3330.

Bates, D., M. Mächler, B. Bolker, and S. Walker, 2015 Fitting linear mixed-effects models using lme4. Journal of Statistical Software **67**: 1–48.

Bernardo, R., 2002 *Breeding for Quantitative Traits in Plants*. Stemma Press.

Burgueño, J., G. de Los Campos, K. Weigel, and J. Crossa, 2012 Genomic Prediction of Breeding Values when Modeling Genotype × Environment Interaction using Pedigree and Dense Molecular Markers. Crop Science **52**: 707.

Calus, M. P. and R. F. Veerkamp, 2011 Accuracy of multi-trait genomic selection using different methods. Genetics Selection Evolution **43**: 26.

Cheng, H., R. Fernando, and D. Garrick, 2018 Jwas: Julia implementation of whole-genome analysis software. In *Proceedings of the World Congress on Genetics Applied to Livestock Production*, volume 11.

Crossa, J., P. Pérez-Rodríguez, J. Cuevas, O. Montesinos-López, D. Jarquín, *et al.*, 2017 Genomic Selection in Plant Breeding: Methods, Models, and Perspectives. Trends in plant science **22**: 961–975.

Daetwyler, H. D., M. P. L. Calus, R. Pong-Wong, G. de Los Campos, and J. M. Hickey, 2013 Genomic Prediction in Animals and Plants: Simulation of Data, Validation, Reporting, and Benchmarking. Genetics **193**: 347–365.

Dahl, A., V. Iotchkova, A. Baud, Å. Johansson, U. Gyllensten, *et al.*, 2016 A multiple-phenotype imputation method for genetic studies. Nature Genetics **48**: 466–472.

de Los Campos, G., J. M. Hickey, R. Pong-Wong, H. D. Daetwyler, and M. P. L. Calus, 2013 Whole-Genome Regression and Prediction Methods Applied to Plant and Animal Breeding. Genetics **193**: 327–345.

Fernandes, S. B., K. O. G. Dias, D. F. Ferreira, and P. J. Brown, 2017 Efficiency of multi-trait, indirect, and trait-assisted genomic selection for improvement of biomass sorghum. TAG Theoretical and applied genetics Theoretische und angewandte Genetik **131**: 747–755.

Friedman, J., T. Hastie, and R. Tibshirani, 2001 *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Gianola, D., 2013 Priors in whole-genome regression: the bayesian alphabet returns. Genetics **194**: 573–596.

Gianola, D. and C. C. Schon, 2016 Cross-Validation Without Doing Cross-Validation in Genome-Enabled Prediction. G3: Genes | Genomes | Genetics **6**: 3107–3128.

Hayes, B. J., P. J. Bowman, A. J. Chamberlain, and M. E. Goddard, 2009 Invited review: Genomic selection in dairy cattle: Progress and challenges. Journal of Dairy Science **92**: 433–443.

Heslot, N., H.-P. Yang, M. E. Sorrells, and J.-L. Jannink, 2012 Genomic Selection in Plant Breeding: A Comparison of Models. Crop Science **52**: 146–160.

Jia, Y. and J.-L. Jannink, 2012 Multiple-Trait Genomic Selection Methods Increase Genetic Value Prediction Accuracy. Genetics **192**: 1513–1522.

Lado, B., D. Vázquez, M. Quincke, P. Silva, I. Aguilar, *et al.*, 2018 Resource allocation optimization with multi-trait genomic prediction for bread wheat (<Emphasis Type="Italic">Triticum aestivum</Emphasis> L.) baking quality. TAG Theoretical and applied genetics Theoretische und angewandte Genetik **131**: 2719–2731.

Legarra, A. and A. Reverter, 2018 Semi-parametric estimates of population accuracy and bias of predictions of breeding values and future phenotypes using the LR method. Genetics Selection Evolution **50**: 659.

Meuwissen, T. H., B. J. Hayes, and M. E. Goddard, 2001 Prediction of total genetic value using genome-wide dense marker maps. Genetics **157**: 1819–1829.

Montesinos-López, O. A., A. Montesinos-López, J. Crossa, D. Gianola, C. M. Hernández-Suárez, *et al.*, 2018 Multi-trait, Multi-environment Deep Learning Modeling for Genomic-Enabled Prediction of Plant Traits. G3: Genes | Genomes | Genetics **8**: g3.200728.2018–3840.

Pszczola, M., R. F. Veerkamp, Y. de Haas, E. Wall, T. Strabel, *et al.*, 2013 Effect of predictor traits on accuracy of genomic breeding values for feed intake based on a limited cow reference population. animal **7**: 1759–1768.

Rutkoski, J., J. Poland, S. Mondal, E. Autrique, L. G. Pérez, *et al.*, 2016 Canopy temperature and vegetation indices from high-throughput phenotyping improve accuracy of pedigree and genomic selection for grain yield in wheat. G3: Genes, Genomes, Genetics **6**: 2799–2808.

Spiliopoulou, A., R. Nagy, M. L. Bermingham, J. E. Huffman, C. Hayward, *et al.*, 2015 Genomic prediction of complex human traits: relatedness, trait architecture and predictive meta-models. Human molecular genetics **24**: 4167–4182.

Thompson, R. and K. Meyer, 1986 A review of theoretical aspects in the estimation of breeding values for multi-trait selection. Livestock Production Science **15**: 299–313.

Utz, H. F., A. E. MELCHINGER, and C. C. Schön, 2000 Bias and Sampling Error of the Estimated Proportion of Genotypic Variance Explained by Quantitative Trait Loci Determined From Experimental Data in Maize Using Cross Validation and Validation With Independent Samples. Genetics **154**: 1839–1849.

Ziyatdinov, A., M. Vazquez-Santiago, H. Brunel, A. Martinez-Perez, H. Aschard, *et al.*, 2018 lme4qtl: linear mixed models with flexible covariance structure for genetic studies of related individuals. BMC Bioinformatics p. btw080.

APPENDIX

Here, we derive the genomic predictions $\hat{\mathbf{u}}_{n1}$ given $\mathbf{y}$ for the three prediction models that we use in the main text, and then evaluate the expected covariances between these predictions and the predictands $\mathbf{u}_{n1}$ and $\mathbf{y}_{n1}$. We derive these relations for the more general situation with $p \geq 1$ "secondary" traits and a single "focal" trait.

We start with a phenotypic data matrix $\mathbf{Y}$ with n individuals and $p+1$ traits, where the first trait (first column of $\mathbf{Y}$) is the "focal" trait, and the other p traits are "secondary" traits. We first divide $\mathbf{Y}$ into a training partition ("old" individuals) and a testing partition ("new" individuals), and arrange them with the testing partition first, so we can partition $\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_n \\ \mathbf{Y}_o \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \mathbf{y}_{n1} & \mathbf{Y}_{n2} \end{bmatrix} \\ \begin{bmatrix} \mathbf{y}_{o1} & \mathbf{Y}_{o2} \end{bmatrix} \end{bmatrix}$. We then work with stacked versions of these phenotype matrices: $\mathbf{y} = vec(\mathbf{Y}), \mathbf{y}_n = vec(\mathbf{Y}_n), \mathbf{y}_o = vec(\mathbf{Y}_o)$. Our genetic model for $\mathbf{y}$ is:

$$
\begin{aligned}
\mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{u} + \mathbf{e} \\
\boldsymbol{\beta} &= [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2]^\intercal \\
\mathbf{u} &\sim \mathrm{N}(\mathbf{0}, \mathbf{G} \otimes \mathbf{K}) \\
\mathbf{e} &\sim \mathrm{N}(\mathbf{0}, \mathbf{R} \otimes \mathbf{I}_n)
\end{aligned}
$$

where $\mathbf{G}$ and $\mathbf{R}$ are genetic and phenotypic covariance matrices for the $p+1$ traits, and $\mathbf{K}$ is the $n \times n$ genomic relationship matrix among the lines. For convenience below, we partition the following matrices as follows: We partition the trait vectors for the training individuals and covariance matrices between the "focal" (index 1) and "secondary traits" (index 2):

$$
\begin{aligned}
\mathbf{y}_o &= \begin{bmatrix} \mathbf{y}_{o1} \\ \mathbf{y}_{o2} \end{bmatrix}, \ \mathbf{u}_o = \begin{bmatrix} \mathbf{u}_{o1} \\ \mathbf{u}_{o2} \end{bmatrix}, \ \mathbf{e}_o = \begin{bmatrix} \mathbf{e}_{o1} \\ \mathbf{e}_{o2} \end{bmatrix}, \ \mathbf{X}_o\boldsymbol{\beta} = \begin{bmatrix} \mathbf{X}_{o1}\boldsymbol{\beta}_1 \\ \mathbf{X}_{o2}\boldsymbol{\beta}_2 \end{bmatrix} \\
\mathbf{G} &= \begin{bmatrix} g_{11} & \mathbf{g}_{12} \\ \mathbf{g}_{21} & \mathbf{G}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{1\cdot} \\ \mathbf{G}_{2\cdot} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{\cdot 1} & \mathbf{G}_{\cdot 2} \end{bmatrix} \\
\mathbf{R} &= \begin{bmatrix} r_{11} & \mathbf{r}_{12} \\ \mathbf{r}_{21} & \mathbf{R}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{1\cdot} \\ \mathbf{R}_{2\cdot} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{\cdot 1} & \mathbf{R}_{\cdot 2} \end{bmatrix},
\end{aligned}
$$

where scalars are normal text, vectors are bold-face lower case letters, and matrices are bold-face capital letters. Partitions for the testing individuals are similar. We also partition the genomic relationship matrix and its inverse between the training and testing individuals:

$$
\mathbf{K} = \begin{bmatrix} \mathbf{K}_{nn} & \mathbf{K}_{no} \\ \mathbf{K}_{on} & \mathbf{K}_{oo} \end{bmatrix}, \ \mathbf{K}^{-1} = \begin{bmatrix} (\mathbf{K}^{-1})_{nn} & (\mathbf{K}^{-1})_{no} \\ (\mathbf{K}^{-1})_{on} & (\mathbf{K}^{-1})_{oo} \end{bmatrix}
$$

Derivation of genomic predictions

Single trait predictions For the single-trait prediction, we begin by estimating $\hat{g}_{11}$, $\hat{r}_{11}$ and $\hat{\boldsymbol{\beta}}_1$ by REML using only $\mathbf{y}_{o1}$. The joint distribution of $\mathbf{u}_{n1}$ and $\mathbf{y}_{o1}$ is:

$$
\begin{bmatrix} \mathbf{u}_{n1} \\ \mathbf{y}_{o1} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{X}_{o1}\boldsymbol{\beta}_1 \end{bmatrix}, \begin{bmatrix} g_{11}\mathbf{K}_{nn} & g_{11}\mathbf{K}_{no} \\ g_{11}\mathbf{K}_{on} & g_{11}\mathbf{K}_{oo} + r_{11}\mathbf{I} \end{bmatrix} \right).
$$

Let: $\mathbf{V}_{o1} = g_{11}\mathbf{K}_{oo} + r_{11}\mathbf{I}$. Therefore $\mathrm{E}[\mathbf{u}_{n1}|\mathbf{y}_{o1}] = g_{11}\mathbf{K}_{no}\mathbf{V}_{o1}^{-1}(\mathbf{y}_{o1} - \mathbf{X}_{o1}\boldsymbol{\beta}_1)$, so our prediction is:

$$
\hat{\mathbf{u}}_{n1}^{(1)} = \hat{g}_{11}\mathbf{K}_{no}\widehat{\mathbf{V}}_{o1}^{-1}(\mathbf{y}_{o1} - \mathbf{X}_{o1}\hat{\boldsymbol{\beta}}_1). \tag{9}
$$

To simplify, note that the joint distribution of $\mathbf{u}_{o1}$ and $\mathbf{y}_{o1}$ in the training data is:

$$
\begin{bmatrix} \mathbf{u}_{o1} \\ \mathbf{y}_{o1} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{X}_{o1}\boldsymbol{\beta}_1 \end{bmatrix}, \begin{bmatrix} g_{11}\mathbf{K}_{oo} & g_{11}\mathbf{K}_{oo} \\ g_{11}\mathbf{K}_{oo} & g_{11}\mathbf{K}_{oo} + r_{11}\mathbf{I} \end{bmatrix} \right)
$$

Therefore, $\hat{\mathbf{u}}_{o1}|\mathbf{y}_{o1} = \hat{g}_{11}\mathbf{K}_{oo}\widehat{\mathbf{V}}_{o1}^{-1}(\mathbf{y}_{o1} - \mathbf{X}_{o1}\hat{\boldsymbol{\beta}}_1)$. Rearranging and plugging this in above simplifies to: $\hat{\mathbf{u}}_{n1}^{(1)} = \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\hat{\mathbf{u}}_{o1}$.

CV1-style multi-trait predictions For CV1-style multi-trait prediction, we begin by estimating $\widehat{\mathbf{G}}$, $\widehat{\mathbf{R}}$ and $\hat{\boldsymbol{\beta}}$ by REML using $\mathbf{y}_o$. The joint distribution of $\mathbf{u}_{n1}$ and $\mathbf{y}_o$ is:

$$\begin{bmatrix} \mathbf{u}_{n1} \\ \mathbf{y}_o \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{X}_o\boldsymbol{\beta} \end{bmatrix}, \begin{bmatrix} g_{11}\mathbf{K}_{nn} & \mathbf{g}_{1\cdot} \otimes \mathbf{K}_{no} \\ \mathbf{g}_{\cdot 1} \otimes \mathbf{K}_{on} & \mathbf{G} \otimes \mathbf{K}_{oo} + \mathbf{R} \otimes \mathbf{I} \end{bmatrix} \right)$$

Let $\mathbf{V}_o = \mathbf{G} \otimes \mathbf{K}_{oo} + \mathbf{R} \otimes \mathbf{I}$. Therefore, $\mathrm{E}[\mathbf{u}_{n1}|\mathbf{y}_o] = (\mathbf{g}_{1\cdot} \otimes \mathbf{K}_{no})\mathbf{V}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\boldsymbol{\beta})$, so our prediction is:

$$\hat{\mathbf{u}}_{n1}^{(2)} = (\hat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\mathbf{V}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}). \tag{10}$$

As above, to simplify this expression, we form the joint distribution of $\mathbf{u}_o$ and $\mathbf{y}_o$ in the training data as:

$$\begin{bmatrix} \mathbf{u}_o \\ \mathbf{y}_o \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{X}_o\boldsymbol{\beta} \end{bmatrix}, \begin{bmatrix} \mathbf{G} \otimes \mathbf{K}_{oo} & \mathbf{G} \otimes \mathbf{K}_{oo} \\ \mathbf{G} \otimes \mathbf{K}_{oo} & \mathbf{G} \otimes \mathbf{K}_{oo} + \mathbf{R} \otimes \mathbf{I} \end{bmatrix} \right)$$

Therefore, $\hat{\mathbf{u}}_{o1}|\mathbf{y}_o = (\hat{\mathbf{G}} \otimes \mathbf{K}_{oo})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}})$. Rearranging and plugging this in above simplifies to: $\hat{\mathbf{u}}_{n1}^{(2)} = \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\hat{\mathbf{u}}_{o1}$.

CV2-style multi-trait predictions For our CV2-style multi-trait prediction, we take a two-step approach. We first estimate $\hat{\mathbf{u}}_o$ from the training individuals and then supplement this with $\mathbf{y}_{n2}$ from the testing individuals. The joint distribution of $\mathbf{u}_{n1}$, $\mathbf{y}_{n2}$ and $\mathbf{u}_o$ is:

$$\begin{bmatrix} \begin{bmatrix} \mathbf{u}_{n1} \\ \mathbf{y}_{n2} \end{bmatrix} \\ \mathbf{u}_o \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{X}_2\boldsymbol{\beta}_2 \end{bmatrix} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{G} \otimes \mathbf{K}_{nn} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \otimes \mathbf{I}_{nn} & \mathbf{G} \otimes \mathbf{K}_{no} \\ \mathbf{G} \otimes \mathbf{K}_{on} & \mathbf{G} \otimes \mathbf{K}_{oo} \end{bmatrix} \right)$$

Conditional on a known value of $\mathbf{u}_o$ from the training individuals, the distribution of $\begin{bmatrix} \mathbf{u}_{n1} \\ \mathbf{y}_{n2} \end{bmatrix}$ would be:

$$\begin{bmatrix} \mathbf{u}_{n1} \\ \mathbf{y}_{n2} \end{bmatrix} |\mathbf{u}_o \sim \mathrm{N}\left(\begin{bmatrix} \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\mathbf{u}_{o1} \\ \mathbf{X}_2\boldsymbol{\beta}_2 + \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\mathbf{u}_{o2} \end{bmatrix}, (\mathbf{G} \otimes \mathbf{K}_{nn}) + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \otimes \mathbf{I}_{nn} - \left[\mathbf{G} \otimes \mathbf{K}_{on})(\mathbf{G}^{-1} \otimes \mathbf{K}_{nn}^{-1})(\mathbf{G} \otimes \mathbf{K}_{no})\right] \right),$$

which simplifies to:

$$\begin{bmatrix} \mathbf{u}_{n1} \\ \mathbf{y}_{n2} \end{bmatrix} |\mathbf{u}_o \sim \mathrm{N}\left(\begin{bmatrix} \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\mathbf{u}_{o1} \\ \mathbf{X}_2\boldsymbol{\beta}_2 + \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\mathbf{u}_{o2} \end{bmatrix}, \begin{bmatrix} g_{11}(\mathbf{K}^{-1})_{nn} & \mathbf{g}_{12} \otimes (\mathbf{K}^{-1})_{nn} \\ \mathbf{g}_{21} \otimes (\mathbf{K}^{-1})_{nn} & \mathbf{G}_{22} \otimes (\mathbf{K}^{-1})_{nn} + \mathbf{R}_{22} \otimes \mathbf{I}_{nn} \end{bmatrix} \right).$$

Let $\mathbf{V}_c = \mathbf{G}_{22} \otimes (\mathbf{K}^{-1})_{nn} + \mathbf{R}_{22} \otimes \mathbf{I}_{nn}$. Now, conditioning on observed values of both $\mathbf{u}_o$ from the training data and $\mathbf{y}_{n2}$ from the testing data, the expectation of $\mathbf{u}_{n1}$ would be:

$$\mathrm{E}[\mathbf{u}_{n1}|\mathbf{y}_{n2}, \mathbf{u}_o] = \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\mathbf{u}_{o1} + (\mathbf{g}_{12} \otimes (\mathbf{K}^{-1})_{nn})\mathbf{V}_c^{-1}(\mathbf{y}_{n2} - \mathbf{X}_2\boldsymbol{\beta}_2 - \mathbf{K}_{oo}\mathbf{K}_{nn}^{-1}\mathbf{u}_{o2}).$$

Using this, we form our prediction as:

$$\hat{\mathbf{u}}_{n1}^{(3)} = \mathbf{K}_{no}\mathbf{K}_{oo}^{-1}\hat{\mathbf{u}}_{o1} + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{y}_{n2} - \mathbf{X}_2\hat{\boldsymbol{\beta}}_2 - \mathbf{K}_{oo}\mathbf{K}_{nn}^{-1}\hat{\mathbf{u}}_{o2}), \tag{11}$$

where $\hat{\mathbf{u}}_{o1}$ and $\hat{\mathbf{u}}_{o2}$ are extracted from the calculation of $\hat{\mathbf{u}}_o$ for the CV1-style prediction. Plugging in the solutions for these values expands to:

$$\begin{aligned} \hat{\mathbf{u}}_{n1}^{(3)} &= (\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}) \\ &\quad + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{y}_{n2} - \mathbf{X}_2\hat{\boldsymbol{\beta}}_2 - (\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}})). \end{aligned}$$

Expectations of prediction accuracy

Now, we evaluate the expected correlation between a random sample of pairs of elements from our three candidate predictions and the predictand $\mathbf{y}_{n1}$. We compare these expected correlations with the expected "true" correlations with $\mathbf{u}_{n1}$. Below, let $var(\mathbf{x})$ denote the variance of a random sample from a random vector $\mathbf{x}$; $cov(\mathbf{x},\mathbf{y})$ and $cor(\mathbf{x},\mathbf{y})$ denote the covariance and correlation between a random sample of pairs of elements from $\mathbf{x}$ and $\mathbf{y}$; and $Cov(\mathbf{x},\mathbf{y})$ denote the covariance matrix between vectors $\mathbf{x}$ and $\mathbf{y}$. We use the following results:

$$cor(\mathbf{x},\mathbf{y}) = \frac{cov(\mathbf{x},\mathbf{y})}{\sqrt{var(\mathbf{x})var(\mathbf{y})}} = \frac{1}{n-1}\frac{(\mathbf{x}-\boldsymbol{\mu}_x)^\intercal(\mathbf{y}-\boldsymbol{\mu}_y)}{\sqrt{var(\mathbf{x})var(\mathbf{y})}} = \frac{1}{n-1}\frac{\mathbf{x}^\intercal\mathbf{S}\mathbf{y}}{\sqrt{var(\mathbf{x})var(\mathbf{y})}}$$

where $\mathbf{S} = \mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\intercal}{n}$.

$$\mathrm{E}[\mathbf{x}^\intercal\mathbf{S}\mathbf{y}] = tr(\mathbf{S}Cov(\mathbf{x},\mathbf{y})) + \mu_x^\intercal\mathbf{S}\mu_y = tr(\mathbf{S}Cov(\mathbf{x},\mathbf{y}))$$

where $tr(\cdot)$ is the matrix trace, and $\boldsymbol{\mu}_x = \mathbf{0}$ and/or $\boldsymbol{\mu}_y = \mathbf{0}$. Therefore, the expected correlation between $\mathbf{x}$ and $\mathbf{y}$ is approximately:

$$\mathrm{E}[cor(\mathbf{x},\mathbf{y})] \approx \frac{1}{n-1}\frac{tr(\mathbf{S}Cov(\mathbf{x},\mathbf{y}))}{\sqrt{\mathrm{E}[var(\mathbf{x})]\mathrm{E}[var(\mathbf{y})]}}.$$

Our goal with cross-validation is to estimate $cor(\hat{\mathbf{u}}_{n1},\mathbf{u}_{n1})$. Since we do not know $\mathbf{u}_{n1}$, we approximate the correlation with $cor(\hat{\mathbf{u}}_{n1},\mathbf{y}_{n1})/\sqrt{\hat{h}_1^2}$. The factor of $\sqrt{\hat{h}_1^2}$ corrects the correlation for the larger variance of $\mathbf{y}_{n1}$ relative to $\mathbf{u}_{n1}$. Otherwise, any difference between these two correlations must be due to their numerators: $tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1},\mathbf{u}_{n1}))$ and $tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1},\mathbf{y}_{n1}))$. Thus, for each of the three prediction methods we compare these two numerators to evaluate the accuracy and bias in the approximation.

Single trait predictions The numerator of the expected correlation between $\mathbf{u}_{n1}^{(1)}$ and the true genetic values $\mathbf{u}_{n1}$ is:

$$\begin{aligned}
tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(1)},\mathbf{u}_{n1})) &= tr\left(\mathbf{S}Cov(\hat{g}_{11}\mathbf{K}_{no}\widehat{\mathbf{V}}_{o1}^{-1}(\mathbf{y}_{o1}-\mathbf{X}_{o1}\hat{\boldsymbol{\beta}}_1),\mathbf{u}_{n1})\right)\\
&= tr\left(\hat{g}_{11}\mathbf{S}\mathbf{K}_{no}\widehat{\mathbf{V}}_{o1}^{-1}Cov(\mathbf{u}_{o1}+\mathbf{e}_{o1},\mathbf{u}_{n1})\right)\\
&= tr\left(\hat{g}_{11}\mathbf{S}\mathbf{K}_{no}\widehat{\mathbf{V}}_{o1}^{-1}(g_{11}\mathbf{K}_{on})\right)\\
&= \hat{g}_{11}g_{11}tr\left(\mathbf{S}\mathbf{K}_{no}\widehat{\mathbf{V}}_{o1}^{-1}\mathbf{K}_{on}\right).
\end{aligned}$$

where we assume that $\hat{\boldsymbol{\beta}}_1 = \boldsymbol{\beta}_1$ and $Cov(\mathbf{e}_{o1},\mathbf{u}_{n1}) = \mathbf{0}$. The same result for the numerator of the expected correlation between $\mathbf{u}_{n1}^{(1)}$ and the observed phenotypic values $\mathbf{y}_{n1}$ is:

$$\begin{aligned}
tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(1)},\mathbf{y}_{n1})) &= tr\left(\mathbf{S}Cov(\hat{g}_{11}\mathbf{K}_{no}\widehat{\mathbf{V}}_{o1}^{-1}(\mathbf{y}_{o1}-\mathbf{X}_{o1}\hat{\boldsymbol{\beta}}_1),\mathbf{y}_{n1})\right)\\
&= tr\left(\hat{g}_{11}\mathbf{S}\mathbf{K}_{no}\widehat{\mathbf{V}}_{o1}^{-1}Cov(\mathbf{u}_{o1}+\mathbf{e}_{o1},\mathbf{u}_{n1}+\mathbf{e}_{n1})\right)\\
&= tr\left(\hat{g}_{11}\mathbf{S}\mathbf{K}_{no}\widehat{\mathbf{V}}_{o1}^{-1}(g_{11}\mathbf{K}_{on})\right)\\
&= \hat{g}_{11}g_{11}tr\left(\mathbf{S}\mathbf{K}_{no}\widehat{\mathbf{V}}_{o1}^{-1}\mathbf{K}_{on}\right),
\end{aligned}$$

where we additionally assume $Cov(\mathbf{u}_{o1},\mathbf{e}_{n1}) = \mathbf{0}$ and $Cov(\mathbf{e}_{o1},\mathbf{e}_{n1}) = \mathbf{0}$. Therefore, the numerators are the same, and $cor(\hat{\mathbf{u}}_{n1}^{(1)},\mathbf{y}_{n1})/\sqrt{\hat{h}_1^2}$ is a consistent estimator for $cor(\hat{\mathbf{u}}_{n1}^{(1)},\mathbf{u}_{n1})$.

CV1-style multi-trait predictions The numerator of the expected correlation between $\mathbf{u}_{n1}^{(2)}$ and the true genetic values $\mathbf{u}_{n1}$ is:

$$\begin{aligned}
tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n}^{(2)},\mathbf{u}_{n1})) &= tr(\mathbf{S}Cov((\hat{\mathbf{g}}_{1\cdot}\otimes\mathbf{K}_{oo})\widehat{\mathbf{V}}_{o}^{-1}(\mathbf{y}_{o}-\mathbf{X}_{o}\hat{\boldsymbol{\beta}}),\mathbf{u}_{n1}))\\
&= tr(\mathbf{S}(\hat{\mathbf{g}}_{1\cdot}\otimes\mathbf{K}_{oo})\widehat{\mathbf{V}}_{o}^{-1}Cov(\mathbf{u}_{o}+\mathbf{e}_{o},\mathbf{u}_{n1}))\\
&= tr(\mathbf{S}(\hat{\mathbf{g}}_{1\cdot}\otimes\mathbf{K}_{oo})\widehat{\mathbf{V}}_{o}^{-1}(\mathbf{g}_{21}\otimes\mathbf{K}_{on})),
\end{aligned}$$

again assuming $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}$ and now also $Cov(\mathbf{e}_{o},\mathbf{u}_{n1}) = \mathbf{0}$. The same result for the numerator of the expected correlation between $\mathbf{u}_{n1}^{(2)}$ and the observed phenotypic values $\mathbf{y}_{n1}$ is:

$$\begin{aligned}
tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(2)},\mathbf{y}_{n1})) &= tr(\mathbf{S}Cov((\hat{\mathbf{g}}_{1\cdot}\otimes\mathbf{K}_{oo})\widehat{\mathbf{V}}_{o}^{-1}(\mathbf{y}_{o}-\mathbf{X}_{o}\hat{\boldsymbol{\beta}}),\mathbf{y}_{n1}))\\
&= tr(\mathbf{S}(\hat{\mathbf{g}}_{1\cdot}\otimes\mathbf{K}_{oo})\widehat{\mathbf{V}}_{o}^{-1}Cov(\mathbf{u}_{o}+\mathbf{e}_{o},\mathbf{u}_{n1}+\mathbf{e}_{n1}))\\
&= tr(\mathbf{S}(\hat{\mathbf{g}}_{1\cdot}\otimes\mathbf{K}_{oo})\widehat{\mathbf{V}}_{o}^{-1}(\mathbf{g}_{21}\otimes\mathbf{K}_{on})),
\end{aligned}$$

where we additionally assume $Cov(\mathbf{u}_{o},\mathbf{e}_{n1}) = \mathbf{0}$ and $Cov(\mathbf{e}_{o},\mathbf{e}_{n1}) = \mathbf{0}$. Therefore, the numerators are the same, and $cor(\hat{\mathbf{u}}_{n1}^{(2)},\mathbf{y}_{n1})/\sqrt{\hat{h}_1^2}$ is a consistent estimator for $cor(\hat{\mathbf{u}}_{n1}^{(2)},\mathbf{u}_{n1})$.

CV2-style multi-trait predictions The numerator of the expected correlation between $\mathbf{u}_{n1}^{(3)}$ and the true genetic values $\mathbf{u}_{n1}$ is:

$$\begin{aligned}
tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{u}_{n1})) &= tr(\mathbf{S}[Cov\Big((\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}) \\
&\qquad - (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}) \\
&\qquad + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{y}_{n2} - \mathbf{X}_2\hat{\boldsymbol{\beta}}_2), \mathbf{u}_{n1}\Big)]) \\
&= tr(\mathbf{S}[Cov((\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}), \mathbf{u}_{n1}) \\
&\quad - Cov((\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}), \mathbf{u}_{n1}) \\
&\quad + Cov((\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{y}_{n2} - \mathbf{X}_2\hat{\boldsymbol{\beta}}_2), \mathbf{u}_{n1})]) \\
&= tr(\mathbf{S}[(\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}Cov(\mathbf{u}_o + \mathbf{e}_o, \mathbf{u}_{n1}) \\
&\quad - (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}Cov(\mathbf{u}_o + \mathbf{e}_o, \mathbf{u}_{n1}) \\
&\quad + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}Cov(\mathbf{u}_{n2} + \mathbf{e}_{n2}, \mathbf{u}_{n1})]) \\
&= tr(\mathbf{S}[(\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on}) \\
&\quad - (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on}) \\
&\quad + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{nn})]) \\
&= tr(\mathbf{S}(\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on})) \\
&\quad - tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on})) \\
&\quad + tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{nn})),
\end{aligned}$$

again assuming $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}$, $Cov(\mathbf{e}_o, \mathbf{u}_{n1}) = \mathbf{0}$, and $Cov(\mathbf{e}_{n2}, \mathbf{u}_{n1}) = \mathbf{0}$. From this, we can see the potential benefit of the CV2-style method:

$$\begin{aligned}
&tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{u}_{n1})) - tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n}^{(2)}, \mathbf{u}_{n1})) \\
&\quad = tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{nn})) - tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on})) \\
&\quad = tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{nn} - (\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on}))),
\end{aligned}$$

which is generally (but maybe not necessarily) positive. This means that $cor(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{u}_{n1})$ is generally greater than $cor(\hat{\mathbf{u}}_{n1}^{(2)}, \mathbf{u}_{n1})$.

The same result for the numerator of the expected correlation between $\mathbf{u}_{n1}^{(3)}$ and the observed phenotypic values $\mathbf{y}_{n1}$ is:

$$\begin{aligned}
tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{y}_{n1})) &= tr(\mathbf{S}[Cov((\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}), \mathbf{u}_{n1} + \mathbf{e}_{n1}) \\
&\quad - Cov((\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}), \mathbf{u}_{n1} + \mathbf{e}_{n1}) \\
&\quad + Cov((\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{y}_{n2} - \mathbf{X}_2\hat{\boldsymbol{\beta}}_2), \mathbf{u}_{n1} + \mathbf{e}_{n1})]) \\
&= tr(\mathbf{S}[(\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}Cov(\mathbf{u}_o + \mathbf{e}_o, \mathbf{u}_{n1} + \mathbf{e}_{n1}) \\
&\quad - (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}Cov(\mathbf{u}_o + \mathbf{e}_o, \mathbf{u}_{n1} + \mathbf{e}_{n1}) \\
&\quad + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}Cov(\mathbf{u}_{n2} + \mathbf{e}_{n2}, \mathbf{u}_{n1} + \mathbf{e}_{n1})]) \\
&= tr(\mathbf{S}[(\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on}) \\
&\quad - (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on}) \\
&\quad + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{nn})]) \\
&\quad + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{r}_{21} \otimes \mathbf{I})]) \\
&= tr(\mathbf{S}(\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on})) \\
&\quad - tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{on})) \\
&\quad + tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{nn})) \\
&\quad + tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{r}_{21} \otimes \mathbf{K}_{nn})),
\end{aligned}$$

From this, we see that the numerator of the correlation $cor(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{y}_{n1})/\sqrt{h_1^2}$ is not equal to that of $cor(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{u}_{n1})$:

$$tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{y}_{n1})) - tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{u}_{n1})) = tr(\mathbf{S}(\hat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{r}_{21} \otimes \mathbf{K}_{nn})).$$

If $p = 1$, then $\hat{\mathbf{g}}_{12}$ and $\mathbf{r}_{12}$ are scalars and this excess covariance is approximately $n\hat{\mathbf{g}}_{12}\mathbf{r}_{12}$.

CV2* approach In our new CV2* cross-validation approach, we replace $\mathbf{y}_{n1}$ with $\mathbf{y}_{x1}$–the phenotypes of a new set of individuals (x) that are relatives of the testing partition and were not part of the training partition. Let $\mathbf{K}_{xx}$ be the genetic relationships among these n_x individuals, and $\mathbf{K}_{xo}$ be their genetic relationships with the training partition. The numerator of the expected correlation $cor(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{y}_{x1})/\sqrt{h_1^2}$ is:

$$
\begin{aligned}
tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{y}_{x1})) &= tr(\mathbf{S}[Cov((\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}), \mathbf{u}_{x1} + \mathbf{e}_{x1}) \\
&\quad - Cov((\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}), \mathbf{u}_{x1} + \mathbf{e}_{x1}) \\
&\quad + Cov((\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{y}_{n2} - \mathbf{X}_2\hat{\boldsymbol{\beta}}_2), \mathbf{u}_{x1} + \mathbf{e}_{x1})]) \\
&= tr(\mathbf{S}[(\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}Cov(\mathbf{u}_o + \mathbf{e}_o, \mathbf{u}_{x1} + \mathbf{e}_{x1}) \\
&\quad - (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}Cov(\mathbf{u}_o + \mathbf{e}_o, \mathbf{u}_{x1} + \mathbf{e}_{x1}) \\
&\quad + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}Cov(\mathbf{u}_{n2} + \mathbf{e}_{n2}, \mathbf{u}_{x1} + \mathbf{e}_{x1})]) \\
&= tr(\mathbf{S}[(\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{ox}) \\
&\quad - (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{ox}) \\
&\quad + (\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{xx})]) \\
&= tr(\mathbf{S}(\widehat{\mathbf{g}}_{1\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{ox})) \\
&\quad - tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\widehat{\mathbf{G}}_{2\cdot} \otimes \mathbf{K}_{no})\widehat{\mathbf{V}}_o^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{ox})) \\
&\quad + tr(\mathbf{S}(\widehat{\mathbf{g}}_{12} \otimes (\mathbf{K}^{-1})_{nn})\widehat{\mathbf{V}}_c^{-1}(\mathbf{g}_{21} \otimes \mathbf{K}_{xx})).
\end{aligned}
$$

If these new individuals are clones of the original testing set, then $\mathbf{K}_{xx} = \mathbf{K}_{nn}$, $\mathbf{K}_{ox} = \mathbf{K}_{on}$ and $tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{y}_{x1})) = tr(\mathbf{S}Cov(\hat{\mathbf{u}}_{n1}^{(3)}, \mathbf{u}_{n1}))$. However, if clones are not available, then this equality will not hold.