High-dimensional sparse vine copula regression with application to genomic prediction

Şahin, Ö.; Czado, Claudia

DOI
10.1093/biomtc/ujad042

Publication date
2023

Document Version
Final published version

Citation (APA)
Şahin, Ö., & Czado, C. (2023). High-dimensional sparse vine copula regression with application to genomic prediction. (1 ed.) https://doi.org/10.1093/biomtc/ujad042

Biometrics, 2024, 80(1), ujad042
https://doi.org/10.1093/biomtc/ujad042
Biometric Methodology

OXFORD

High-dimensional sparse vine copula regression with application to genomic prediction

Özge Sahin[1,2,*] and Claudia Czado[1,3]

[1]Department of Mathematics, Technical University of Munich, Boltzmannstraße 3, 85748 Garching, Germany, [2]Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands, [3]Munich Data Science Institute, Walther-von-Dyck-Straße 10, 85748 Garching, Germany

[*]Corresponding author: Özge Sahin, Department of Mathematics, Technical University of Munich, 85748 Garching, Germany (O.Sahin@tudelft.nl).

ABSTRACT

High-dimensional data sets are often available in genome-enabled predictions. Such data sets include nonlinear relationships with complex dependence structures. For such situations, vine copula-based (quantile) regression is an important tool. However, the current vine copula-based regression approaches do not scale up to high and ultra-high dimensions. To perform high-dimensional sparse vine copula-based regression, we propose 2 methods. First, we show their superiority regarding computational complexity over the existing methods. Second, we define relevant, irrelevant, and redundant explanatory variables for quantile regression. Then, we show our method's power in selecting relevant variables and prediction accuracy in high-dimensional sparse data sets via simulation studies. Next, we apply the proposed methods to the high-dimensional real data, aiming at the genomic prediction of maize traits. Some data processing and feature extraction steps for the real data are further discussed. Finally, we show the advantage of our methods over linear models and quantile regression forests in simulation studies and real data applications.

KEYWORDS: genomic prediction; high-dimensional data; quantile regression; variable selection; vine copula.

1 INTRODUCTION

Genomic prediction (GP) aims at predicting a breeding value using genotypic measurements. Then, an unobserved trait is predicted using its genotype information like single-nucleotide polymorphism (SNP). With rapid developments in genomic technologies, researchers have high-dimensional SNP data sets. However, it poses some challenges in prediction modeling, such as a small number of observations and a large number of explanatory variables, skewness in variables, irrelevant and redundant variables, interactions among variables, and nonconstant error variance. To solve the drawbacks regarding the data dimensionality in the GP, statistical or machine learning-based approaches have been applied (Li et al., 2018). Recently, quantile regression approaches, which model the conditional distribution of the response, have been utilized to deal with the skewness and outliers in the data (Pérez-Rodríguez et al., 2020). Still, the question has been *how to model conditional quantiles flexibly while handling data dimensionality in GP*.

It is important to *identify the SNPs relevant* for predicting breeding values to design future genotype studies. For instance, since the human population has been growing, the stability of food supplies has gained much more importance. Thus, plant breeding efforts aim at the crops' genetic improvement. Hölker et al. (2019) provided agronomic measurements and >500 000 SNPs to make European flint maize landraces available for such an aim.

Vine copula-based (quantile) regression allows modeling a nonlinear relationship between explanatory variables and responses. It considers higher-order explanatory variables and deals with unknown functional error forms. However, the current vine copula-based regression methods' computational complexity makes them infeasible to be applied in high-dimensional data sets (Kraus and Czado, 2017; Tepegjozova et al., 2022). We refer to high- and ultra-high-dimensional data sets when the number of explanatory variables is between 10 and 1000 and >1000, respectively.

We propose 2 vine copula-based regression methods that perform well in analyzing high-dimensional sparse data sets, where sparsity means that many explanatory variables do not predict the response. Their computational complexity is significantly less than the existing methods. We define relevant, redundant, and irrelevant explanatory variables for quantile regression and assess the methods' prediction power in high-dimensional sparse simulated data sets. Our analyses regarding the inclusion of relevant variables and exclusion of irrelevant variables show our methods' capability to provide sparse models. We apply the methods for genomic prediction of maize traits, proposing data preprocessing, and feature extraction steps on the data given by Hölker et al. (2019). Such steps can be further applied and improved in future studies. Overall, we can resolve data dimensionality issues in vine copula-based regression. To the best of our knowledge, there has not yet been any study performing the ge-

Received: August 25, 2022; Revised: October 5, 2023; Accepted: December 29, 2023

nomic prediction using vine copula models and assessing vine copula regression methods' performance in the presence of redundant and irrelevant variables.

Alternative nonlinear quantile regression models are generalized additive models (*GAM*) (Wood, 2017), quantile regression forests (*QRF*) (Meinshausen, 2006), and quantile regression neural networks (*QRNN*) (Cannon, 2011). *QRNN* may suffer from quantile crossing (Cannon, 2018), which does not exist in vine copula-based approaches by construction. Kraus and Czado (2017) show a better performance of their vine copula-based approach than *GAM*. Hence, among nonlinear models, we compare our models with quantile regression forests and show our advantages, especially in the presence of dependent variables. Moreover, despite the quantile crossing problem, we analyze the performance of linear models with variable selection, ie, linear quantile regression with a LASSO-type penalty (*LQRLasso*) (Belloni and Chernozhukov, 2011), in nonlinear cases.

The paper is organized as follows: Section 2 introduces vine copulas and new methods; Section 3 provides simulation studies. We present the real data application in Section 4, discuss our findings, and conclude in Section 5. The paper has online Supplementary Material.

2 HIGH-DIMENSIONAL SPARSE VINE COPULA REGRESSION

2.1 D-vine copulas and prediction

Copulas are distribution functions, allowing us to separate the univariate margins and dependence structure. Let $\boldsymbol{X} = (X_1, \dots, X_p)^\top \in \mathbb{R}^p$ be a p-dimensional random vector with the joint cumulative distribution function (cdf) F and the univariate marginal distributions $F_1, \dots F_p$. By Sklar's theorem (Sklar, 1959), the copula C, corresponding to F, is a multivariate cdf with uniform margins such that $F(x_1, \dots, x_p) = C\big[F_1(x_1), \dots F_p(x_p)\big]$. When the univariate marginal distributions are continuous, C is unique, which we assume in the remainder. In addition, the p-dimensional joint density f can be written as $f(\boldsymbol{x}) = c\big[F_1(x_1), \dots, F_p(x_p)\big] \times f_1(x_1) \cdots f_p(x_p)$, $\boldsymbol{x} \in \mathbb{R}^p$, where c is the copula density of the random vector $\big[F_1(X_1), \dots F_p(X_p)\big]^\top \in [0, 1]^p$.

Standard multivariate copulas, such as the multivariate exchangeable Archimedean or Gaussian, often do not accurately model the dependence among the variables. Aas et al. (2009) developed a pair-copula construction or vine copula approach using a cascade of bivariate copulas to extend their flexibilities. Such a construction can be represented by an undirected graphical structure involving a set of linked trees, ie, a regular (R-) vine (Bedford and Cooke, 2002). A R-vine for p variables consists of $p - 1$ trees, where the edges in the first tree are the nodes of the second tree, the edges of the second tree are the nodes of the third tree, and so on. If an edge connects 2 nodes in the $(t+1)$th tree, their associated edges in the tth tree must have a shared node in the tth tree for $t = 1, \dots, p-2$. The nodes and edges in the first tree represent p variables and unconditional dependence for $p - 1$ pairs of variables, respectively. In the higher trees, the conditional dependence of a pair of variables conditioning on other variables is modeled. To get a vine copula or pair-copula construction, there is a bivariate (pair) copula associated with each edge in the vine.

One special class of R-vines are D-vines, whose graph structure is a path, ie, all nodes' degree in the graph is smaller than three. A node in a path represents a variable, and an edge between a pair of nodes corresponds to dependence among the variables of the respective nodes expressed by a pair copula. The node having a degree of one is called a leaf node. Once the order of the nodes in the first tree is determined, the associated D-vine copula decomposition is unique. Moreover, if the pair copulas in the higher tree levels than t are independence copulas, where $t < p$ and $t \geq 1$, representing conditional independence, a t-truncated vine copula is obtained (Section 1 of the Supplementary Material).

D-vine copulas allow us to express the conditional density of a leaf node in the first tree in a closed form. Here, we choose the leaf node in the first tree of a D-vine copula as a response variable. In the remainder, assume that $(y_i, x_{i,1}, \dots, x_{i,p})$, $i = 1, \dots, n$, are realizations of the random vector $(Y, X_1, \dots, X_p)$, and Y denotes a response variable with its marginal distributions F_Y and the others correspond to explanatory variables with their marginal distributions $F_1, \dots, F_p$. For the D-vine copula with the node order $0 - 1 - \dots - p$ corresponding to the variables $Y - X_1 - \dots - X_p$, $p \geq 2$, as stated in Kraus and Czado (2017), the conditional quantile function $F^{-1}_{0|1,\dots,p}$ at quantile α can be expressed in terms of the inverse marginal distribution function F_Y^{-1} of the response and the conditional D-vine copula quantile function $C^{-1}_{0|1,\dots,p}$ at quantile α as $F^{-1}_{0|1,\dots,p}(\alpha|x_1, \dots, x_p) = F_Y^{-1}\big[C^{-1}_{0|1,\dots,p}(\alpha|F_1(x_1), \dots, F_p(x_p))\big]$.

Since p-dimensional D-vine copula's input is the marginally uniform data on $[0, 1]^p$, the estimation of the D-vine copula follows a two-step approach called the inference for margins (Joe and Xu, 1996). First, each marginal distribution is estimated. Then, the data is converted into the copula data by applying probability integral transformation, eg, using a univariate non-parametric kernel density estimator, ie, $\hat{F}_Y$ and $\hat{F}_{X_d}$, $d = 1, \dots, p$. Next, we have the pseudo copula data: $(v_i, u_{i,1}, \dots, u_{i,p}) = [\hat{F}_Y(y_i), \hat{F}_{X_1}(x_{i,1}), \dots, \hat{F}_{X_p}(x_{i,p})]$, $i = 1, \dots, n$, being realizations of the random vector $(V, U_1, \dots, U_p)$. More details about the conditional log-likelihood and estimation of D-vine copulas are in Section 1 of the Supplementary Material.

2.2 Proposed methods: *vineregRes* and *vineregParCor*

We propose 2 methods to perform a D-vine copula regression on high-dimensional sparse data sets: *vineregRes* and *vineregParCor*. Sections 2 and 3 of the Supplementary Material give an illustrative example and details of the methods.

The method *vineregRes* performs the variable selection at a given iteration based on the residuals of the previous iteration, ie, the pseudo-response. It finds the variable among the candidates that provides the best bivariate copula conditional log-likelihood conditioned on the variable and conditioning the pseudo-response of the previous iteration. Assume $\tilde{y}_i^{(s)}$ and $\tilde{v}_i^{(s)}$ $i = 1, \dots, n$, denote the pseudo-response and its pseudo-copula data in the sth iteration, respectively, which are realizations of the random variable $Y^{(s)}$ and $V^{(s)}$, respectively. $V^{(0)}$ and

$V^{(s)}$ have the indices 0 and $0^{(s)}$, respectively, and are always a leaf node in the first tree.

vineregRes

Step 1 (initialization): For given data $(y_i, x_{i,1}, \ldots, x_{i,p})$ and $(v_i, u_{i,1}, \ldots, u_{i,p})$, define the initial pseudo-response $\tilde{y}_i^{(1)} = y_i$ with its copula scale $\tilde{v}_i^{(1)}$, $i = 1, \ldots, n$, the initial D-vine order $\mathcal{D}^{(1)} = (0)$, the initial chosen variable index set $\mathcal{I}_{var}^{(1)} = \emptyset$, and the initial set of candidate explanatory variables $p_{cand}^{(1)} = \{1, \ldots, p\}$.

For $s = 1, 2, \ldots,$

Step 2 (variable selection): Fit a parametric bivariate copula to data $\{(\tilde{v}_i^{(s)}, u_{i,d}), i = 1, \ldots, n\}$ for $d \in p_{cand}^{(s)}$ and denote the copula, copula density, and its estimated parameters by $\widehat{CR}^{d_{(s)}}$, $\widehat{cr}^{d_{(s)}}$ and $\hat{\boldsymbol{\theta}}^{d_{(s)}}$, respectively. Then, find the variable for which the conditional log-likelihood of the copula $\widehat{CR}^{d^*_{(s)}}$ is maximized, ie,

$d^*_{(s+1)} = \arg\max_{d_{(s)} \in p_{cand}^{(s)}} \sum_{i=1}^{n} \ln \widehat{cr}^{d_{(s)}}_{0^{(s)}|d_{(s)}} [\tilde{v}_i^{(s)} | u_{i,d_{(s)}}; \hat{\boldsymbol{\theta}}^{d_{(s)}}].$

Step 3 (D-vine extension): Extend the D-vine order by adding the variable with index $d^*_{(s+1)}$ to get a D-vine order $\mathcal{D}^{(s+1)} = [\mathcal{D}^{(s)}, d^*_{(s+1)}]$. Select the parametric pair copula families and estimate the parameters in the extended D-vine structure, where the associated D-vine copula and its estimated parameters are denoted by $\hat{C}^{(s+1)}$ and $\hat{\boldsymbol{\theta}}^{(s+1)}$, respectively.

Step 4 (chosen variable indices and hyperparameter updates): Extend the chosen variable set, adding the new variable, $\mathcal{I}_{var}^{(s+1)} = \mathcal{I}_{var}^{(s)} \cup d^*_{(s+1)}$ and update $p_{cand}^{(s+1)} = p_{cand}^{(s)} \setminus d^*_{(s+1)}$.

Step 5 (pseudo-response update or stop): If a stopping condition (Section 2.3) does not hold, estimate the median of the response variable based on the D-vine copula $\hat{C}^{(s+1)}$ and update the pseudo-response, ie,

$\tilde{y}_i^{(s+1)} = y_i - \hat{F}_Y^{-1}[\hat{C}^{-1(s+1)}_{0|\boldsymbol{I}_{var}^{(s+1)}}(0.50 | u_{i,p_1}, \ldots, u_{i,p_{d_{(s+1)}}}; \hat{\boldsymbol{\theta}}^{(s+1)})]$,
$\tilde{v}_i^{(s+1)} = \hat{F}_{Y^{(s+1)}}[\tilde{y}_i^{(s+1)}]$,

where $i = 1, \ldots, n$ and $\{p_1, \ldots, p_{d_{(s+1)}}\} \subseteq \mathcal{I}_{var}^{(s+1)}$.

Another method to perform a D-vine copula regression for high-dimensional data is to use the partial correlation between the response and a candidate explanatory variable given the chosen variables at each iteration based on their empirical normal scores (Joe, 2014).

vineregParCor

Step 1 (initialization): As given in *vineregRes* and the data's normal scores.

For $s = 1, 2, \ldots,$

Step 2 (variable selection): $d^*_{(s+1)} = \arg\max_{d_{(s)} \in p_{cand}^{(s)}} |\hat{\rho}_{0, d_{(s)}; \mathcal{I}_{var}^{(s)}}|$, where $\hat{\rho}_{j,k;S}$ is the estimated partial correlation of variables j, k given those indexed in S based normal scores.

Step 3 (D-vine extension): As given in *vineregRes*.

Step 4 (chosen variable indices and hyperparameter updates): As given in *vineregRes*.

2.3 Bivariate copula selection and stopping criteria

Step 3 of *vineregRes* and *vineregParCor* selects parametric pair copulas and estimates their parameters associated with the extension of the D-vine structure. Step 2 of *vineregRes* fits a parametric bivariate copula to the pseudo-response and a candidate explanatory variable. First, we estimate the parameters that maximize the log-likelihood of a candidate bivariate copula family (Section 4 of the Supplementary Material). Later, we can select the one with the lowest Akaike (AIC) or the Bayesian information criterion. While extending the D-vine structure and adding new trees at Step 3 of the methods, the fit of parametric pair copulas can be performed sequentially from the lowest to the highest trees (Brechmann, 2010).

To decide if a chosen candidate explanatory variable in a given iteration should be in a model, we will consider the conditional AIC (cAIC), which penalizes the conditional log-likelihood of the model based on the D-vine copula by the effective degrees of freedom in the model (Section 4 of the Supplementary Material). We stop adding variables when the current iteration's (cAIC) is not smaller than the previous iteration's (cAIC). If the cAIC always improves in each iteration, we stop after all explanatory variables are included in the model.

2.4 Complexity

Assuming that the data consists of p explanatory variables, the complexity of the existing method *vinereg* is $\mathcal{O}(p^3)$ in terms of the total number of bivariate copulas to be selected during the algorithm (Tepegjozova, 2019). Thus, we evaluate the complexity of *vineregRes* and *vineregParCor* using the same criterion. We will consider the worst-case scenario that the algorithms run until all explanatory variables are included in the model. Further, the total number of estimated parameters is linear in terms of the number of bivariate copulas. The detailed calculations in Section 4 of the Supplementary Material show that the complexity of *vineregParCor* and *vineregRes* in terms of the total number of selected bivariate copulas is $\mathcal{O}(p^2)$. Hence, our methods significantly reduce the computational complexity of *vinereg*.

2.5 Relevant, redundant, and irrelevant variables

Now we define relevant, redundant, and irrelevant variables for predicting the conditional quantile of a response variable Y given the index set of explanatory variables $\mathcal{X}$. We will denote the cdf of the variables with the index set $\mathcal{X}$ by $F_{\mathcal{X}}$ in the following.

Definition 1 *(Relevant variables) The index set of variables $\mathcal{M} \subseteq \mathcal{X}$ is called relevant for Y if and only if it holds $F_{Y|\mathcal{M}}(y|\boldsymbol{x}_{\mathcal{M}}) \neq F_Y(y)$, where $\boldsymbol{x}_{\mathcal{M}}$ includes the variables in $\mathcal{M}$.*

Definition 2 *(Redundant variables) The index set of variables $\mathcal{R}$ is called redundant given the set of variables $\mathcal{M}$ for Y if and only if it holds $F_{Y|\mathcal{M},\mathcal{R}}(y|\boldsymbol{x}_{\mathcal{M}}, \boldsymbol{x}_{\mathcal{R}}) = F_{Y|\mathcal{M}}(y|\boldsymbol{x}_{\mathcal{M}})$ and $F_{\mathcal{M},\mathcal{R}}(\boldsymbol{x}_{\mathcal{M}}, \boldsymbol{x}_{\mathcal{R}}) \neq F_{\mathcal{M}}(\boldsymbol{x}_{\mathcal{M}}) \times F_{\mathcal{R}}(\boldsymbol{x}_{\mathcal{R}})$, where the vectors $\boldsymbol{x}_{\mathcal{M}}$ and $\boldsymbol{x}_{\mathcal{R}}$ include the variables in the sets $\mathcal{M}$ and $\mathcal{R}$, respectively, $\mathcal{R} \subseteq \mathcal{X}$, $\mathcal{M} \subseteq \mathcal{X}$, $\mathcal{R} \cap \mathcal{M} = \emptyset$.*

A discussion on redundant variables in a D-vine copula is in Section 5 of the Supplementary Material.

Example 1 *Consider the model* $(Y, X_1, X_2)^\top \sim \mathcal{N}_3(\mathbf{0}, \Sigma)$ *with (0.5, 0.4, 0.8) vectorizing the upper triangular part of the symmetric covariance matrix* Σ*, where it holds* $\rho_{Y,X_2;X_1} = 0$*, ie,* Y *is conditionally independent of* X_2 *given* X_1*. Hence, we have* $f_{Y|X_1,X_2}(y|x_1, x_2) = \frac{f_{Y,X_2|X_1}(y,x_2|x_1)}{f_{X_2|X_1}(x_2|x_1)} = \frac{f_{Y|X_1}(y|x_1)\times f_{X_2|X_1}(x_2|x_1)}{f_{X_2|X_1}(x_2|x_1)} = f_{Y|X_1}(y|x_1)$*. Since* $f_{X_1,X_2}(x_1, x_2) \neq f_{X_1}(x_1) \times f_{X_2}(x_2)$*,* X_2 *is redundant given* X_1 *for* Y*.*

Definition 3 *(Irrelevant variables) The set of variables* $\mathcal{I}$ *is called irrelevant given the set of variables* $\mathcal{M}$ *for* Y *if and only if it holds* $F_{Y|\mathcal{M},\mathcal{I}}(y|\boldsymbol{x}_\mathcal{M}, \boldsymbol{x}_\mathcal{I}) = F_{Y|\mathcal{M}}(y|\boldsymbol{x}_\mathcal{M})$*,* $F_{\mathcal{M},\mathcal{I}}(\boldsymbol{x}_\mathcal{M}, \boldsymbol{x}_\mathcal{I}) = F_\mathcal{M}(\boldsymbol{x}_\mathcal{M}) \times F_\mathcal{I}(\boldsymbol{x}_\mathcal{I})$*, and* $F_{Y,\mathcal{I}}(y, \boldsymbol{x}_\mathcal{I}) = F_Y(y) \times F_\mathcal{I}(\boldsymbol{x}_\mathcal{I})$*, where the vectors* $\boldsymbol{x}_\mathcal{M}$ *and* $\boldsymbol{x}_\mathcal{I}$ *include the variables in the sets* $\mathcal{M}$ *and* $\mathcal{I}$*, respectively,* $\mathcal{I} \subseteq \mathcal{X}$*,* $\mathcal{M} \subseteq \mathcal{X}$*,* $\mathcal{I} \cap \mathcal{M} = \emptyset$*.*

Example 2 *Consider the model* $(Y, X_1, X_2)^\top \sim \mathcal{N}_3(\mathbf{0}, \Sigma)$ *with (0.5, 0, 0) vectorizing the upper triangular part of the symmetric covariance matrix* Σ*, where it holds* $\rho_{Y,X_2;X_1} = 0$*; hence,* $f_{Y|X_1,X_2}(y|x_1, x_2) = f_{Y|X_1}(y|x_1)$*. In addition, it holds* $f_{X_1,X_2}(x_1, x_2) = f_{X_1}(x_1) \times f_{X_2}(x_2)$ *and* $f_{Y,X_2}(y, x_2) = f_Y(y) \times f_{X_2}(x_2)$*; thus,* X_2 *is irrelevant given* X_1 *for* Y*.*

3 SIMULATION STUDY

We show the flexibility and effectiveness of the proposed methods on simulated datasets being nonlinear and having different sparsity. We explore the following questions: Q1 How do *vineregRes* and *vineregParCor* work in situations with nonlinear explanatory variable effects on the response's quantiles in the presence of redundant and irrelevant variables for prediction accuracy and computational complexity? Q2 How well do *vineregRes* and *vineregParCor* identify relevant and irrelevant variables for predicting the response's quantiles? Q3 How do *vineregRes* and *vineregParCor* perform compared to the alternative methods *LQRLasso* and *QRF* (Section 6 of the Supplementary Material)?

3.1 Data generating process (DGP)

DGP1: irrelevant variables

$$\begin{aligned} Y_i^d &= X_{i,1} \times X_{i,2}^2 \times \sqrt{|X_{i,3}| + 0.1} + e^{0.4\times X_{i,4}\times X_{i,5}} \\ &\quad + (X_{i,6}, \ldots, X_{i,p_d})(0, \ldots, 0)^\top \\ &\quad + \epsilon_i \times \sigma_i, \quad i = 1, \ldots, n, \quad d = 1, 2, 3, \qquad (1) \end{aligned}$$

where we sample the relevant variables $(X_{i,1}, \ldots, X_{i,5})^\top \sim \mathcal{N}_5(\mathbf{0}, \Sigma)$, $i = 1, \ldots, n$ with the (a, b)th element of the covariance matrix $\Sigma_{a,b} = 0.75^{|a-b|}$, irrelevant variables $(X_{i,6}, \ldots, X_{i,p_d})^\top \sim \mathcal{N}_{p_d-5}(\mathbf{0}, \mathbb{I}_{p_d-5})$, the random error terms $\epsilon_i \sim \mathcal{N}(0, 1)$ that are independent and identically distributed (iid), independently, and set $\sigma_i \in \{0.5, 1\}$, $i = 1, \ldots, n$. We simulate data sets with different number of irrelevant variables and set it to $(p_d - 5)$ in each case $d = 1, 2, 3$ as follows: Case 1 with $p_1 = 10$ (50% of variables are irrelevant), Case 2 with $p_2 = 20$ (75% of variables are irrelevant), and Case 3 with $p_3 = 50$ (90% of variables are irrelevant).

DGP2: redundant variables

$$\begin{aligned} Y_i^d &= \sqrt{|5 \times X_{i,1} - 2 \times X_{i,9} + 0.5|} + X_{i,8} \times (-4 \times X_{i,3} + 1) \\ &\quad + e^{X_{i,6}} + (2 \times X_{i,10}^3 + X_{i,4}^3) \\ &\quad + (X_{i,7} + 1) \times (\ln(|X_{i,2} + X_{i,5}| + 0.01)) \\ &\quad + (X_{i,11}, \ldots, X_{i,p_d})(0, \ldots, 0)^\top \\ &\quad + \epsilon_i \times \sigma_i, \quad i = 1, \ldots, n, \quad d = 1, 2, 3, 4, \qquad (2) \end{aligned}$$

where the samples of explanatory variables are independently generated from a multivariate normal distribution with a Toeplitz correlation structure, ie, $(X_{i,1}, \ldots, X_{i,p_d})^\top \sim \mathcal{N}_{p_d}(\mathbf{0}, \Sigma)$, $i = 1, \ldots, n$, $j = 1, 2, 3$, with the (a, b)th element of the covariance matrix $\Sigma_{a,b} = \rho^{|a-b|}$. To represent a challenging but realistic scenario, we set $\rho = 0.75$. We sample $\epsilon_i \sim \mathcal{N}(0, 1)$, $i = 1, \ldots, n$ (iid) independently from the explanatory variables and set $\sigma_i \in \{0.5, 1\}$, $i = 1, \ldots, n$. All variables predict the response's quantiles; however, the others are redundant, given the first 10 variables. We change the number of redundant variables and set it to $(p_d - 10)$ for $d = 1, 2, 3, 4$: Case 1 with $p_1 = 20$ (50% of variables are redundant given the 10 relevant ones), Case 2 with $p_2 = 40$ (75% of variables are redundant given the 10 relevant ones), Case 3 with $p_3 = 100$ (90% of variables are redundant given the 10 relevant ones), Case 4 with $p_4 = 1000$ (99% of variables are redundant given the 10 relevant ones).

Based on Equations 1 and 2, we simulate samples with size of 450 (n= 450) with a random split of 300/150 observations for a training/a test set. We replicate our procedure 100 times and average performance measures per sample (Section 6 of Supplementary Material).

3.2 Performance measures

We consider the computation time, the number of chosen variables, *true positive rate (TPR)*, and *false discovery rate (FDR)* as methods' performance measures on the training set. To evaluate the performance of a method on the test set, we apply *the pinball loss* (PL_α) at $\alpha = 0.05, 0.50, 0.95$. TPR is the ratio of the chosen relevant variables by a method M to the total number of relevant variables. FDR is the ratio of the number of chosen irrelevant variables to the total number of chosen variables by a method M. Higher TPR and smaller FDR are better. PL_α measures the accuracy of the quantile predictions $\hat{y}_i^{\alpha,M}$ at the level α by a method M compared to the given response y_i, $i = 1, \ldots, n$. The smaller pinball loss values are better (Steinwart and Christmann, 2011) (Section 6 of the Supplementary Material).

3.3 Results

All computations are run on a single-node CPU with Intel Xeon Platinum 8380H Processor with ~25 GB RAM, running R version 4.2.2. However, Step 2 of *vineregRes* can be broken down into parallel fits across candidate variables for a faster computation.

Variable selection and computational complexity results on the training set: In Table 1, we analyze the TPR and FDR only for the DGP1 setting since all variables are relevant in the DGP2 setting. The computations for *vinereg* were not complete within 3 days

TABLE 1 Comparison of the methods' performance on the training set over 100 replications under the cases 1–3 and 1–4 specified in Equations 1 and 2, respectively.

DGP	Measure	Case	*vinereg*	*vineregRes*	*vineregParCor*	*QRF*	*LQRLasso* (0.05, 0.50, 0.95)
1	TPR	1	0.81 (0.01)	0.80 (0.02)	0.67 (0.02)	1.00 (0.00)	0.73 (0.02), 0.69 (0.02), 0.63 (0.02)
		2	0.85 (0.01)	0.79 (0.03)	0.59 (0.02)	1.00 (0.00)	0.68 (0.02), 0.68 (0.02), 0.55 (0.02)
		3	0.89 (0.01)	0.78 (0.02)	0.56 (0.02)	1.00 (0.00)	0.61 (0.02), 0.61 (0.02), 0.48 (0.01)
	FDR	1	0.28 (0.01)	0.08 (0.01)	0.24 (0.02)	0.50 (0.00)	0.28 (0.02), 0.32 (0.02), 0.24 (0.02)
		2	0.55 (0.01)	0.13 (0.02)	0.45 (0.02)	0.75 (0.00)	0.38 (0.02), 0.49 (0.03), 0.34 (0.03)
		3	0.80 (0.00)	0.15 (0.02)	0.65 (0.02)	0.90 (0.00)	0.35 (0.03), 0.58 (0.02), 0.40 (0.03)
	Chosen Vars.	1	5.83 (0.13)	4.56 (0.19)	4.68 (0.16)	10.00 (0.00)	5.24 (0.22), 5.48 (0.21), 4.99 (0.26)
		2	9.76 (0.19)	5.04 (0.26)	6.03 (0.23)	20.00 (0.00)	6.60 (0.41), 8.48 (0.43), 5.30 (0.36)
		3	23.24 (0.42)	5.29 (0.33)	8.74 (0.30)	50.00 (0.00)	5.93 (0.38), 10.20 (0.68), 5.97 (0.47)
	Time	1	0.18 (0.00)	0.17 (0.01)	0.06 (0.00)	0.01 (0.00)	0.02 (0.00), 0.02 (0.00), 0.02 (0.00)
		2	0.97 (0.02)	0.30 (0.02)	0.10 (0.01)	0.01 (0.00)	0.02 (0.00), 0.43 (0.03), 0.02 (0.00)
		3	12.06 (0.31)	0.77 (0.05)	0.34 (0.03)	0.03 (0.00)	0.12 (0.00), 0.14 (0.00), 0.12 (0.00)
2	Chosen Vars.	1	10.94 (0.23)	5.41 (0.19)	6.68 (0.20)	20.00 (0.00)	7.05 (0.32), 10.12 (0.36), 8.95 (0.40)
		2	19.63 (0.54)	5.17 (0.17)	8.25 (0.28)	40.00 (0.00)	7.88 (0.43), 12.36 (0.51), 9.88 (0.48)
		3	62.92 (2.78)	5.83 (0.22)	11.66 (0.42)	100.00 (0.00)	7.35 (0.36), 15.45 (0.78), 9.44 (0.55)
		4	–	7.08 (0.54)	31.72 (1.23)	1000.00 (0.00)	–
	Time	1	1.15 (0.03)	0.20 (0.01)	0.16 (0.01)	0.01 (0.00)	0.09 (0.00), 0.10 (0.00), 0.08 (0.00)
		2	7.30 (0.26)	0.32 (0.01)	0.29 (0.03)	0.02 (0.00)	0.11 (0.00), 0.13 (0.00), 0.11 (0.00)
		3	159.22 (8.66)	0.84 (0.03)	0.72 (0.06)	0.05 (0.00)	0.35 (0.00), 0.35 (0.00), 0.34 (0.00)
		4	–	9.44 (0.69)	12.18 (1.26)	0.44 (0.00)	–

The numbers in parentheses under a method's name column are the corresponding empirical standard errors. (–) shows computational infeasibility. *LQRLasso* column corresponds to the quantile levels (0.05, 0.50, and 0.95). Chosen Vars. corresponds to the total number of chosen variables. Time is in minutes and per replication.

per replication for the fourth case of the DGP2 setting, making it computationally infeasible. Also, we did not run *LQRLasso* for that case since it ran ∼7 h per replication and had worse performances in the other simulation cases.

In all cases, *QRF* chooses all variables in the associated DGP to make predictions. Thus, its TPR is 1, the number of selected variables equals the total number of variables in a sample, and its FDR is the proportion of the irrelevant variables in the associated DGP setting. More analyses about *QRF* are provided in Section 7 of the Supplementary Material.

Excluding *QRF*, in all cases of the DGP1 setting, *vinereg* has a better TPR performance than the others. However, its FDR is higher than others, adding many irrelevant variables to the model. *vineregRes* correctly identifies >75% of the relevant variables in all cases of the DGP1 setting. Its FDR is <15% there, making it the best method for FDR. Further, *vineregParCor*'s TPR is higher than 50% in all DGP1 cases. However, like others, its FDR increases as the number of irrelevant variables increases in the model, reaching >50% in the third DGP1 case. *LQRLasso* identifies at least 48% of the relevant variables, but its TPR decreases when the number of irrelevant variables increases.

While *vineregRes* selects the lowest number of variables between 4 and 6, *vinereg* includes almost half of the total number of variables in the data in each case. This highlights the power of *vineregRes* regarding the exclusion of irrelevant variables in sparse data sets. *vineregParCor*'s number of chosen variables is between *vinereg* and *vineregRes* in all evaluated cases. The same applies to *LQRLasso*, but it selects more variables for estimating median predictions than other quantiles. In an ultra-high-dimensional case with 1000 explanatory variables, the number of variables chosen by *vineregParCor* is, on average, 31.72 with the empirical standard error of 1.23, while it is 7.08 with that of 0.54 for *vineregRes*.

As the number of variables increases, the average running time for all methods increases. Among vine-based methods, *vineregParCor* provides the fastest computation as expected from the results in Section 2.4. However, *QRF* provides the fastest computation among all methods considered. *vineregRes* and *vineregParCor* run <15 min in the ultra-high-dimensional case. *LQRLasso*'s running time for quantile levels does not differ much.

Prediction accuracy results on the test set: Table 2 shows that *vineregRes* provides the best fit in 8 evaluations out of 9 (3 pinball losses evaluated for 3 levels) in the DGP1 setting among vine copula-based methods. *vinereg* and *vineregParCor* have the same accuracy as *vineregRes* for the first case in the DGP1 setting. However, as the number of irrelevant variables increases, a residual-based variable selection may be better than other vine copula-based methods. Moreover, *LQRLasso* has the highest pinball loss in all cases of the DGP1 setting because of the high nonlinearity in samples. Even though *vineregParCor*'s performance is better than *LQRLasso*, it provides worse fits than the others at the level 95%. A likely explanation can be that including irrelevant variables in addition to the most relevant ones in a vine copula may negatively impact the prediction accuracy. However, a similar result does not apply to *QRF*. Despite including all irrelevant variables in the model, *QRF* still performs better than all in 7 evaluations out of 9.

Table 2 shows that *vineregRes* provides the lowest pinball loss at 3 quantile levels in all DGP2 cases, except at the level of 95% in the first case. Since *vineregRes* gives the most sparse models in the DGP2 setting in Table 1, we infer that including many relevant but potentially redundant variables in *vinereg*, *vineregParCor*, and *QRF* is worsening the prediction accuracy in the DGP2 setting. *LQRLasso* suffers from nonlinearity.

TABLE 2 Comparison of the average performance of the methods on the test set for the pinball loss (PL_α) at different quantile levels α over 100 replications under the cases 1–3 and 1–4 specified in Equations 1 and 2, respectively.

DGP	Measure	Case	*vinereg*	*vineregRes*	*vineregParCor*	*QRF*	*LQRLasso*
1	$PL_{0.05}$	1	**0.21 (0.01)**	**0.21 (0.01)**	**0.21 (0.01)**	0.22 (0.01)	0.34 (0.01)
		2	0.23 (0.01)	**0.22 (0.01)**	**0.22 (0.01)**	**0.22 (0.01)**	0.34 (0.02)
		3	0.24 (0.01)	**0.21 (0.01)**	0.22 (0.01)	0.22 (0.01)	0.32 (0.01)
	$PL_{0.50}$	1	0.79 (0.04)	0.79 (0.03)	0.81 (0.04)	**0.76 (0.04)**	0.94 (0.02)
		2	0.84 (0.04)	0.79 (0.03)	0.82 (0.04)	**0.69 (0.02)**	0.97 (0.04)
		3	0.84 (0.02)	0.76 (0.02)	0.79 (0.02)	**0.72 (0.02)**	0.90 (0.02)
	$PL_{0.95}$	1	0.43 (0.07)	0.43 (0.06)	0.46 (0.07)	**0.41 (0.07)**	0.59 (0.04)
		2	0.37 (0.04)	0.39 (0.04)	0.44 (0.04)	**0.32 (0.03)**	0.64 (0.07)
		3	0.38 (0.03)	0.38 (0.03)	0.41 (0.03)	**0.35 (0.03)**	0.53 (0.03)
2	$PL_{0.05}$	1	**0.53 (0.01)**	**0.53 (0.01)**	0.54 (0.01)	0.70 (0.02)	0.84 (0.02)
		2	0.57 (0.01)	**0.54 (0.01)**	0.56 (0.01)	0.70 (0.02)	0.85 (0.02)
		3	0.81 (0.04)	**0.54 (0.01)**	0.58 (0.01)	0.72 (0.01)	0.92 (0.03)
		4	–	**0.55 (0.01)**	0.89 (0.02)	0.78 (0.02)	–
	$PL_{0.50}$	1	1.87 (0.02)	**1.83 (0.02)**	1.84 (0.02)	1.91 (0.02)	2.20 (0.02)
		2	1.99 (0.02)	**1.84 (0.02)**	1.86 (0.02)	1.93 (0.03)	2.26 (0.03)
		3	2.59 (0.09)	**1.84 (0.02)**	1.94 (0.02)	2.00 (0.03)	2.29 (0.02)
		4	–	**1.89 (0.03)**	2.42 (0.03)	2.17 (0.03)	–
	$PL_{0.95}$	1	**0.53 (0.01)**	0.55 (0.01)	0.55 (0.01)	0.65 (0.01)	0.78 (0.02)
		2	0.57 (0.01)	**0.56 (0.01)**	**0.56 (0.01)**	0.67 (0.01)	0.82 (0.02)
		3	0.81 (0.05)	**0.57 (0.01)**	0.61 (0.02)	0.68 (0.01)	0.83 (0.02)
		4	–	**0.57 (0.02)**	0.88 (0.03)	0.75 (0.02)	–

The best performance for each quantile level and DGP case is highlighted. The numbers in parentheses under a method's name column are the corresponding empirical standard errors. (—) shows computational infeasibility.

Since the relevant, irrelevant, and redundant variables are known in simulation studies, when only the relevant 10 variables are used for prediction in the DGP2 setting, *QRF* has the pinball loss of 0.64, 1.81, and 0.62 at levels 0.05, 0.50, and 0.95, respectively. Thus, *vineregRes* would have better accuracy than *QRF* in most cases of the DGP2 setting, even if the latter selected the most relevant variables. Thus, *vineregRes* is more advantageous than *QRF* in the presence of many dependent variables in our simulations.

4 APPLICATION: THE GENOMIC PREDICTION OF MAIZE TRAITS

We describe a real-data application on the doubled-haploid (DH) lines from European flint maize landraces that motivates our methods' usage. Hölker et al. (2019) evaluated 899 DH lines whose data contains genotypic measurements with the SNP array technology and phenotypic measurements of agronomic traits across environments.

We are interested in the relationship between a DH line's genotype encoded by its SNPs and its phenotypic outcome described by its traits, ie, the genomic prediction of maize traits. Specifically, we would like to find relevant SNPs for a trait in a multivariate prediction model using our high-dimensional vine copula regression methods, performing variable selection.

4.1 Data description and preprocessing

There are 3 landraces in the data, and we focus on the Kemater Landmais Gelb (KE) landrace, which has the largest number of observations (471 out of 899). There are 501 124 explanatory variables, SNPs, which have only 0 and 2 as values; eg, 0 corresponds to the genotype TT, and 2 denotes the genotype CC. We predict 4 responses of agronomic traits separately: early plant height measured by centimeters at the fourth and sixth stages (PH_V4/V6), female flowering time (FF), and male flowering time (MF) measured by days (Figure 1). Plant breeders need to increase the early plant development and avoid decreasing or increasing female and male flowering times during the maize genotype adoption. Thus, the traits' prediction from the genotypic measurements is crucial.

To compare the performance of regression methods, we partition our data randomly into training (67%) and test (33%) sets. Then the former and latter contain 314 and 157 observations, respectively. Further, we remove the duplicate explanatory variables, retaining one and the common explanatory variables with the threshold of 5%. For instance, assume an explanatory variable in our training set contains 300 zero values and 14 two values. Then, such a variable does not differ among the observations and might not be expected to have predictive power on a response. Thus, the number of explanatory variables in the training and test sets decreases from 501 124 to 44 789, ie, we retain ∼9% of the initial explanatory variables. Hence, the number of observations (DH lines) in the training sets is 314, whereas 147 for their test sets. The number of explanatory variables (SNPs) is 44 789 ($p = 1, \dots, 44789$), and there are 4 univariate responses (traits) ($k = 1, \dots, 4$).

4.2 Feature extraction

Since our explanatory variables are binary, and there can be associated latent variables with a prediction power on the response,

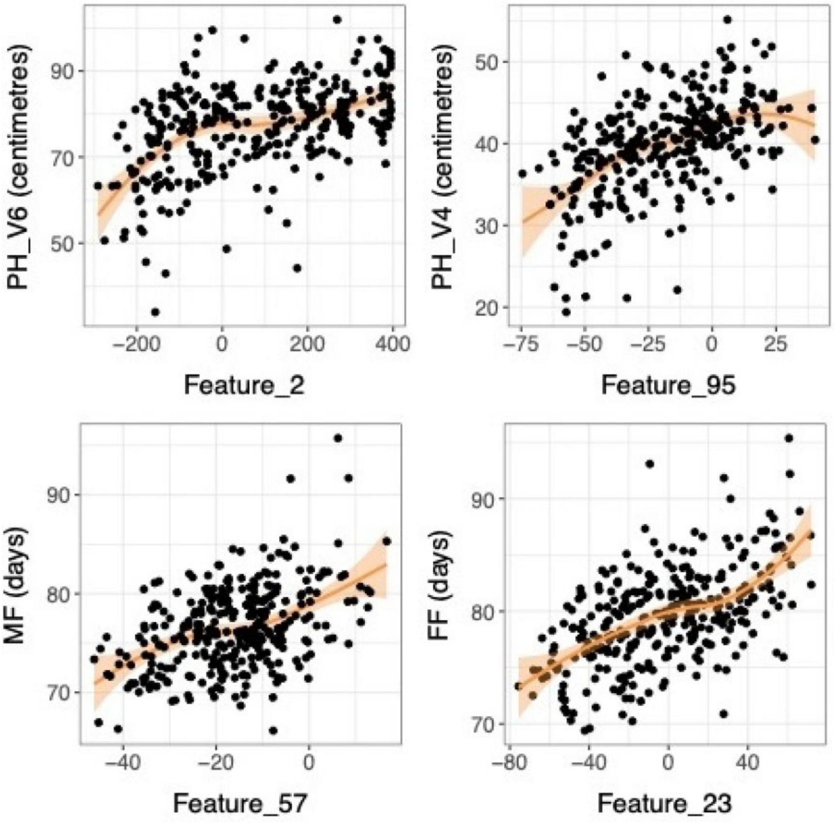

FIGURE 1 Scatter plots of an extracted continuous feature, combining 100 SNPs in a feature, and the traits. Feature_2 corresponds to the combination of the SNPs whose *P*-value from the OLS of the associated trait is higher than 100 SNPs but lower than others. Similar correspondence applies to other features. Orange curves demonstrate a local polynomial regression fit.

we focus on estimating these latent variables and using them as extracted features in a regression method. Our approach is to group the explanatory variables and estimate their weights in their groups so that such weights are used to estimate the latent variables representing each group. Let $\boldsymbol{y}_k$ and $\boldsymbol{SNP}_p$ denote the response vector for trait k and explanatory variable vector p, respectively.

1. Fit a linear regression between a response and an explanatory variable, SNP:
   $\boldsymbol{y}_k = \hat{\beta}_0{}^{p,k} + \hat{\beta}_1{}^{p,k} \times \boldsymbol{SNP}_p, \quad for \quad k = 1, \ldots, 4, \quad p = 1, \ldots, 44789.$
2. Perform a two-tailed Wald test for $H_0 : \beta_1^{p,k} = 0$ versus $H_1 : \beta_1^{p,k} \neq 0$ and determine the associated P-values $P^{p,k}$ for $k = 1, \ldots, 4, \quad p = 1, \ldots, 44789.$
3. Screen the explanatory variables whose P-value from the 2nd step are smaller than 0.10 and have the screened set: $S_k = \{\boldsymbol{SNP_p} : P^{p,k} < 0.10 | p = 1, \ldots, 44789\}$ for $k = 1, \ldots, 4.$
4. Order the set of the explanatory variables S_k based on their P-value non-decreasingly:
   $O_k = \{\boldsymbol{SNP_{w_1}}, \ldots, \boldsymbol{SNP_{w_{|S_k|}}}\}$ with $P^{w_1,k} \leq \ldots \leq P^{w_{|S_k|},k}$ for $O_k = S_k, k = 1, \ldots, 4.$
5. Estimate the latent variables, ie, create the continuous features $\boldsymbol{feature}_{k,d_k}^G$ by using a grouping size G of explanatory variables in O_k and using their coefficients from Equation 1:
   $\boldsymbol{feature}_{k,d_k}^G = \hat{\beta}_1{}^{w_{d_k},k} \times \boldsymbol{SNP_{w_{d_k}}} + \ldots + \hat{\beta}_1{}^{w_{d_k+G-1},k} \times \boldsymbol{SNP_{w_{d_k+G-1}}} \quad \text{for} \quad G \in \{100, 200\},$
   $n_{k_G} = \left\lceil \frac{|O_k|}{G} \right\rceil, \quad d_k = 1, \ldots, n_{k_G}, \quad k = 1, \ldots, 4.$

Then we have 174 (87) continuous features for FF, 92 (46) continuous features for MF, 198 (99) continuous features for PH_V4, and 183 (93) continuous features for PH_V6 by grouping $G = 100$ ($G = 200$). Figure 1 shows a scatter plot of a continuous feature and a trait.

4.3 Prediction

We have our data $D_k^G = (\boldsymbol{y}_k, \boldsymbol{feature}_{k,1}^G, \ldots, \boldsymbol{feature}_{k,n_{k_G}}^G)$ for each response $k = 1, \ldots, 4$ and $G \in \{100, 200\}$. To identify if a feature is relevant, redundant, or irrelevant, we first conduct a bivariate analysis by fitting a vine copula regression on each feature and trait, ie, D-vines with 2 nodes: response and 1 feature. If a feature is relevant or redundant given the others, our methods add it to the model; otherwise, it is not selected as explained in Section 2.5. The bivariate copula family selection between the response and the first feature is conducted as explained in Section 2.3. For instance, we conduct 174 (87) bivariate analyses for FF using a grouping size of $G = 100$ (200). Then all features of 4 responses are classified as relevant or redundant using a grouping

TABLE 3 Comparison of the methods' performance on the test set for the pinball loss (PL_α) and on the training set for the number of selected continuous features (No. Ftr.), where (a, b, and c) under the *LQRLasso* column corresponds to the quantile levels (0.05, 0.50, and 0.95).

		vregRes	*vregParCor*	*LQRLasso*	*QRF*	*vregRes*	*vregParCor*	*LQRLasso*	*QRF*
Trait	**Measure**	**$G = 100$**				**$G = 200$**			
FF	$PL_{0.05}$	**0.35**	0.49	0.40	0.38	0.39	0.39	0.39	**0.37**
	$PL_{0.50}$	**1.43**	1.51	1.50	1.48	1.48	1.56	1.47	**1.45**
	$PL_{0.95}$	0.47	0.47	**0.41**	0.38	0.41	0.43	**0.39**	**0.39**
	No. Ftr.	11	22	(8, 41, 4)	174	4	14	(8, 29, 5)	87
MF	$PL_{0.05}$	0.35	0.36	0.34	**0.33**	0.35	0.36	**0.32**	0.34
	$PL_{0.50}$	1.41	1.42	1.39	**1.36**	1.39	1.40	**1.36**	1.37
	$PL_{0.95}$	0.45	0.47	0.41	**0.39**	0.44	0.45	0.40	**0.39**
	No. Ftr.	12	16	(7, 45, 8)	92	8	13	(5, 15, 12)	46
PH_V4	$PL_{0.05}$	**0.51**	**0.51**	0.55	0.55	**0.51**	0.55	0.56	0.55
	$PL_{0.50}$	1.93	**1.87**	1.92	1.94	1.96	1.99	**1.92**	1.94
	$PL_{0.95}$	0.56	0.58	**0.55**	0.60	0.57	0.57	**0.55**	0.62
	No. Ftr.	6	11	(9, 15, 8)	198	3	11	(7, 17, 4)	99
PH_V6	$PL_{0.05}$	1.01	1.01	**0.98**	1.00	**0.96**	0.98	1.05	1.00
	$PL_{0.50}$	3.09	3.10	**3.04**	3.27	**3.06**	3.47	3.14	3.31
	$PL_{0.95}$	0.91	**0.89**	0.92	0.97	**0.90**	1.05	0.94	1.04
	No. Ftr.	4	12	(8, 49, 5)	183	2	12	(6, 29, 6)	93

The best performance on the test set for each quantile level α, trait, and G is highlighted.

size of $G = 100$ and $G = 200$. Next, we apply our methods to 8 different data sets' training sets to find the most relevant features, thereby redundant ones given them. Also, we compare them with *LQRLasso* and *QRF* on test sets using the pinball loss defined in Section 3.2 at the levels 0.05, 0.50, 0.95.

Table 3 shows that vine copula-based methods perform worse than *LQRLasso* and *QRF* for MF. Dependencies among MF and its selected features by *vineregRes* are more linear than those among other traits since it fits mostly the Gaussian copula in the first tree for MF (Section 8 of the Supplementary Material). We remark that *LQRLasso* may perform well if it can avoid crossing quantile curves, but there is no guarantee that the 95% quantile curve exceeds the 90% quantile curve everywhere (Section 9 of the Supplementary Material). Whenever *LQRLasso* is more accurate than *vineregRes* for PH_V4, it includes more features, giving a trade-off between model sparsity and accuracy. Even though *QRF* provides the lowest pinball loss at all quantiles for FF for $G = 200$, *vineregRes* has better performance than it for $G = 100$, except at the level 95%. *vineregRes* is the most sparse and accurate model at all quantile levels considered for PH_V6 using $G = 200$. It chooses 2 features for $G = 200$, identifying >95% of the features as redundant. It has the best accuracy for PH_V6 for 4 cases out of 6, with 3 quantile levels evaluated for 2 G values.

Given the selected features, the others are redundant for a trait and a grouping of G using *vineregRes*. For instance, given the 1st and 88th features for PH_V6 using a grouping size of $G = 200$, the remaining 91 features are redundant using *vineregRes*. Since the features of PH_V6 are highly dependent but are not needed in a model, in parallel to the simulation study results in Section 3.3, the reason for our methods' better accuracy than *QRF* may be many dependent but redundant features for PH_V6 (Section 10 of the Supplementary Material).

Our SNP screening and feature extraction steps are similar to Qian et al. (2020). Even though they fit a simple linear regression on the first feature, which is based on the linearly and marginally most important SNPs, we conclude in Section 10 in the Supplementary Material that the linearly and marginally most important SNP group might not be considered the most relevant for prediction when allowing nonlinear dependencies as in our methods.

5 DISCUSSION AND CONCLUSION

High-dimensional sparse vine copula regression is a significant tool for efficiently allowing nonlinear relationships between explanatory variables and responses and selecting relevant variables. In genomic prediction, genotypic measurements like SNPs are often very high-dimensional, which might be reduced by considering some SNP groups and their interactions. Also, many groups may be irrelevant for prediction. Our methods can handle such situations and predict responses at different quantile levels. Their performance might be improved with bivariate copula families having more asymmetries, eg, >2 parameters.

For our application, consider the following question: which SNPs impact the low and high quantiles of the trait PH_V6? *vineregRes* identifies 2 SNP groups (features) that consist of 400 SNPs in total. In the first feature, the corresponding SNPs' *P*-values out of the linear regression with the trait PH_V6 are in the range $[10^{-12}, 10^{-7}]$, whereas its range is $[0.087, 0.090]$ in the 88th feature. Thus, the marginal impacts of the selected SNPs differ. Given these SNPs, others are redundant to predict the trait PH_V6. Thus, plant breeders can assess the selected SNP groups' impact on the trait's various quantile levels and identify the associated SNPs using our methods. Chosen SNP groups can be compared to other genome-wide association studies, helping breeders to decide on future genotype adoption. Hence, comparing the identified SNPs with those in Mayer et al. (2020) is high on the agenda.

Feature extraction is a vital step that may impact our methods' genomic prediction power. For instance, the choice of SNPs' weights for estimating their latent variable is open to future research. Also, even though it offers a trade-off between a computational burden and prediction power, one can apply cross-validation for the choice of the SNP's group size G. In addition, some SNPs might affect the trait, not marginally only in the presence of certain other SNPs. Alternatively, one may remove the P-value screening of the SNPs at the 10% level described in Section 4.1 and consider all possible extracted features. Likewise, some SNPs might influence the trait marginally, but not when certain other SNPs are in the model. For such cases, some post-processing steps for feature extraction might be applied.

Finally, our variable selection steps can be adapted for more flexible vine tree structures.

ACKNOWLEDGMENTS

We thank Chris-Carolin Schön for funding acquisition, Harry Joe for helpful suggestions, and Manfred Mayer for useful comments on the data. We are grateful to anonymous referees and the associate editor for comments leading to a considerably improved contribution.

SUPPLEMENTARY MATERIALS

Supplementary material is available at *Biometrics* online.

Additional figures, tables, information, and codes referenced in Sections 2.1, 2.2, 2.3, 2.5, 3, 3.1, and 4.3 are available with this paper at the Biometrics website on Oxford Academic. An R package, called `sparsevinereg`, containing the implementation for *vineregRes* and *vineregParCor* is given on GitHub: https://github.com/oezgesahin/sparsevinereg.

FUNDING

The Munich Data Science Institute funds this research through Seed Funding 2021. C.C. acknowledges the support of the German Research Foundation (DFG grant CZ 86/6-1).

CONFLICT OF INTEREST

None declared.

DATA AVAILABILITY

The data used in this paper to illustrate our findings are given by the paper "Discovery of beneficial haplotypes for complex traits in maize landraces" at https://doi.org/10.1038/s41467-020-18683-3 and openly available at https://figshare.com/articles/journal_contribution/Data_from_Mayer_et_al_2020_Nat_Commun_/12137142.

REFERENCES

Aas, K., Czado, C., Frigessi, A. and Bakken, H. (2009) Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44, 182–198.

Bedford, T. and Cooke, R. M. (2002) Vines—a new graphical model for dependent random variables. *The Annals of Statistics*, 30, 1031–1068.

Belloni, A. and Chernozhukov, V. (2011) L1-penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39, 82–130.

Brechmann, E. (2010) Truncated and simplified regular vines and their applications. Msc thesis, Technical University of Munich.

Cannon, A. J. (2011) Quantile regression neural networks: implementation in R and application to precipitation downscaling. *Computers & Geosciences*, 37, 1277–1284.

Cannon, A. J. (2018) Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. *Stochastic Environmental Research and Risk Assessment*, 32, 3207–3225.

Hölker, A. C., Mayer, M., Presterl, T., Bolduan, T., Bauer, E., Ordas, B. et al. (2019) European maize landraces made accessible for plant breeding and genome-based studies. *Theoretical and Applied Genetics*, 132, 3333–3345.

Joe, H. (2014) *Dependence Modeling with Copulas*. Boca Raton, FL: CRC Press.

Joe, H. and Xu, J. J. (1996) The estimation method of inference functions for margins for multivariate models. Technical report no. 166, Department of Statistics, University of British Columbia.

Kraus, D. and Czado, C. (2017) D-vine copula based quantile regression. *Computational Statistics & Data Analysis*, 110, 1–18.

Li, B., Zhang, N., Wang, Y. G., George, A. W., Reverter, A. and Li, Y. (2018) Genomic prediction of breeding values using a subset of SNPs identified by three machine learning methods. *Frontiers in Genetics*, 9, 237.

Mayer, M., Hölker, A. C., González-Segovia, E., Bauer, E., Presterl, T., Ouzunova, M. et al. (2020) Discovery of beneficial haplotypes for complex traits in maize landraces. *Nature Communications*, 11, 1–10.

Meinshausen, N. (2006) Quantile regression forests. *Journal of Machine Learning Research*, 7, 983–999.

Pérez-Rodríguez, P., Montesinos-López, O. A., Montesinos-López, A. and Crossa, J. (2020) Bayesian regularized quantile regression: a robust alternative for genome-based prediction of skewed data. *The Crop Journal*, 8, 713–722.

Qian, J., Tanigawa, Y., Du, W., Aguirre, M., Chang, C., Tibshirani, R. et al. (2020) A fast and scalable framework for large-scale and ultrahigh-dimensional sparse regression with application to the UK Biobank. *PLoS Genetics*, 16, e1009141.

Sklar, M. (1959) Fonctions de repartition an dimensions et leurs marges. *Publications de l'Institut de statistique de l'Université de Paris*, 8, 229–231.

Steinwart, I. and Christmann, A. (2011) Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17, 211–225.

Tepegjozova, M. (2019) D-and C-vine quantile regression for large data sets. Msc thesis, Technical University of Munich.

Tepegjozova, M., Zhou, J., Claeskens, G. and Czado, C. (2022) Nonparametric C-and D-vine-based quantile regression. *Dependence Modeling*, 10, 1–21.

Wood, S. N. (2017) *Generalized Additive Models: An Introduction with R*. Boca Raton, FL: Chapman and Hall/CRC.

Received: August 25, 2022; Revised: October 5, 2023; Accepted: December 29, 2023