## SUPPLEMENTARY INFORMATION

**Table S1: MLP and CNN hyperparameters together with value ranges optimized with a genetic algorithm.**

| Hyperparameter | Values | Utility | Potential issues |
|---|---|---|---|
| Number of layers | [1, 2, 3, 4, 5] | The bigger the number, the higher the flexibility. | It increases the number of weights to learn, increasing computing time and memory cost. |
| Neurons per layer | [16, 32, 64, 128] | The bigger the number, the higher the flexibility. | It increases the number of weights to learn, increasing the time and memory cost. |
| Convolutional kernel width* | [2, 3, 5, 10] | A larger kernel allows learning more complex patterns. | A smaller kernel has the risk to underfit while a large one suffers the risk of overfitting. |
| Optimizer | ['rmsprop', 'adam', 'sgd', 'adagrad', 'adadelta', 'adamax', 'nadam'] | Optimization algorithms for training deep models includes some specializations to solve different challenges | There is no consensus about which algorithm performs the best. |
| Activation | ['relu', 'elu', 'tanh', 'sigmoid', 'hard_sigmoid', 'softplus', 'linear'] | Makes possible to learn non-linear complex functional mappings between the inputs and response variable. | There is no best activation function. |
| Weight regularization | [0.01, 0.15, 0.225, 0.3] | Decreasing the weight regularization allows the model to fit the training data better. | Low value may cause overfitting. |
| Dropout | [0.01, 0.15, 0.225, 0.3] | A higher dropout helps to reduce overfitting. | Information is lost when nodes are ignored by a high dropout. |

* Defined only in CNNs, it is the number of SNPs per window.

**Table S2: Correlation in height prediction for several SNP window sizes and strides.**

| Window size | Stride | BEST | UNIF |
|---|---|---|---|
| 2 | 1 | 0.336 | 0.263 |
| 3 | 1 | 0.432 | 0.249 |
| 5 | 1 | 0.414 | 0.259 |
| 10 | 1 | 0.405 | 0.269 |
| 2 | 2 | 0.425 | 0.240 |
| 3 | 3 | 0.418 | 0.245 |
| 5 | 5 | 0.406 | 0.254 |
| 10 | 10 | 0.391 | 0.234 |

Results shown were obtained with CNN3 method and 10k BEST SNP set.

**Algorithm 1**

**procedure** GA($p$, $\beta$, $\delta$)
  Generate $p$ feasible solutions randomly $\rightarrow Pop$ ▷ $\rightarrow$ is the assign function
  **for** i=1 to $\delta$ **do** ▷ $\delta$ number of generations
    **for all** members of $Pop$ **do**
      score based on fitness score ▷ R value in validation
    **end for**
    $Sel = \{\}$
    $n_e = n \cdot \beta$ ▷ $n_e$ number of elitism
    $Pop_2 = \{\text{Best } n_e \text{ solutions in } Pop\}$ ▷ Select the best $n_e$ solutions in $Pop$
    $Sel = Sel \cup Pop_2$
    $Pop_2 = \{n_e \text{ random solutions from } Pop \setminus Sel\}$ ▷ Select other $n_e$ solutions
    $Sel = Sel \cup Pop_2$
    Mutate some of the parameters on some of the networks in $Sel$
    $Pop = Sel$
    $n_c = n - |Sel|$
    **for** j=1 to $n_c$ **do** ▷ Breeding between $Sel$
      Randomly select two solutions from $Sel$: $(X_A, X_B)$
      Combine a random assortment of parameters from $X_A$ and $X_B \rightarrow X_C$
      $Pop = Pop \cup X_C$
    **end for**
  **end for** ▷ Each iteration through these steps is called a generation
  **return** Pop
**end procedure**

**Figure S1: Genetic algorithm used for hyperparameter optimization.** The genetic algorithm is defined with the following parameters: p, number of individuals (hyperparameter combinations) per generation; β, percentage of individuals selected and δ, number of generations.

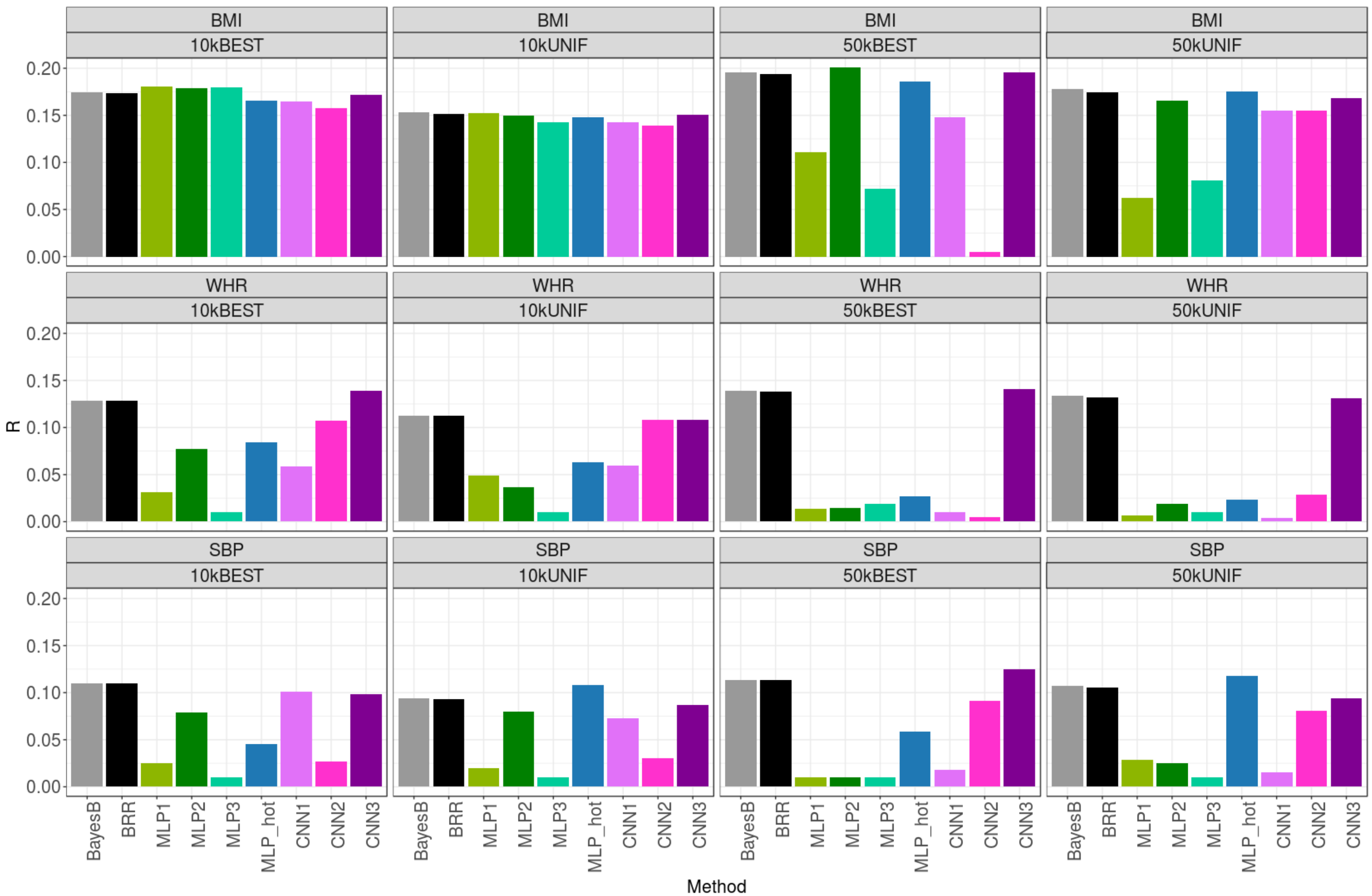


**Figure S2: Prediction performance across methods and traits.** Grey, green, blue and magenta bars correspond to linear, MLPs, hot-encoding MLP and CNNs methods, respectively. Very low bar means method not converging. Average SE of R's were ~$4\times10^{-3}$.