Transformers for Genomic Prediction

María Inés Fariello[1][0000−0002−3337−1209], Graciana Castro[2], Romina Hoffman[2], Mateo Musitelli[1] Diego Belzarena[2], and Federico Lecumberry[2][0000−0002−5491−2019]

[1] Instituto de Matemática y Estadística, Facultad de Ingeniería, Universidad de la República, J. Herrera y Reissig 565, Montevideo, 11300, Uruguay
[2] Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de la República, J. Herrera y Reissig 565, Montevideo, 11300, Uruguay
{fariello, gcastro, romina.hoffman, mmusitelli, dbelzarena, lecumberry}@fing.edu.uy
https://iie.fing.edu.uy/

Abstract. AI is becoming state-of-the-art across scientific fields, giving novel solutions to age-old problems. In genomic prediction, Machine Learning methods could not outperform linear regressions in a general way yet, but are becoming closer. An important feature when working with genomic data, which is non other than a long sequence of information, is to account for the linkage disequilibrium, i.e. dependencies between genome variations that do not need to be close in the genome, and variate with respect to the reference genome. To explode this feature, we evaluate a *Transformer* trained in a small yeast dataset. Although it did not outperform the state-of-the-art results yet, the model got close achieving an R^2 score of 0.389 and 0.400 in Lactate and Lactose ambients, respectively, comparing to the R^2 score of 0.568 and 0.582 for Lactate and Lactose ambients, for the linear model of Lasso, proposed by [7].This proves that there is still room for improvement.

Keywords: Genomic Prediction · SNPs · genotype · phenotype · Neural Networks · Transformers.

1 Introduction

Genomic prediction involves using the information contained in the genome of an individual or a population to make inferences about phenotypes, such as specific traits or diseases. It is based on the premise that certain variations in DNA, such as nucleotide variations known as Single Nucleotide Polymorphisms (SNPs) are associated, due to linkage disequilibrium, with mutations responsible for the variation that certain traits present, or the presence or absence of diseases. In this context, improving the interpretation and prediction of data is a constant challenge due to significant differences in data sets, population structure, and sample size.

To continuously improve the results of linear models and seek alternatives to these models, we aim to apply *Transformers* [13] in the field of genomic

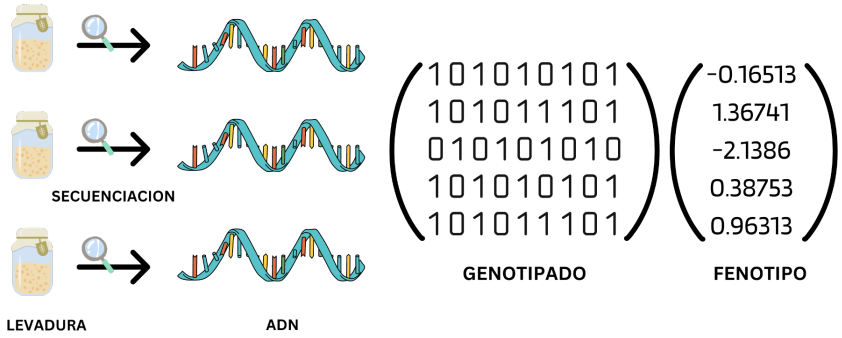

Fig. 1. Representation of the input data to our problem.

Prediction, as they have demonstrated great capacity for capturing long-term relationships in sequences. Successfully adapting and training a model that can extract and learn biological dependencies from dependencies between positions in data sequences could lead to a major breakthrough [3]. The ability of *Transformers* to capture contextual information and model long-range dependencies makes them strong candidates for this task.

We propose to train a model based on the one proposed by Jubair et al. [8] to predict yeast growth in two different environments, Lactate and Lactose.

2 Problem Description

We have a database with information on yeast growth in forty-eight different environments. The yeast database contains growth information for 1,008 yeast strains in forty-eight different environments. Each strain includes information on 11,623 SNPs, encoded with values zero or one depending on whether the individual presents a variation at that position in their genotype. The phenotype value that quantifies its growth in that environment is associated with each individual.

The problem addressed is predicting yeast growth in each of the aforementioned environments. Specifically, we worked with the Lactate and Lactose environments. Yeast growth is a phenotype that is quantified numerically, having for each yeast genotype its corresponding growth phenotype, as illustrated in Figure 1.

Although genomic prediction is a very promising approach in the field of genetics, increasing the *accuracy* of genomic predictions across various models remains a challenge. Multi-phenotypic models, that is, those that predict multiple phenotypes simultaneously, have shown promising results when evaluated according to the article *"Multi-trait multi-environment genomic prediction of agronomic traits in advanced breeding lines of winter wheat"* [5]. In light of the aforementioned, we therefore implement a multi-trait Transformer model and seek to compare its results to those of a single-trait Transformer model.

Additionally, a commonly used approach in multivariate genetics is index selection, which assigns different weights to each trait based on its economic importance. However, classical index selection only optimizes genetic gain in the next generation and requires experimentation to find the weights that lead to the desired outcomes, according to the article *"Multi-trait genomic selection methods for crop improvement"* [10].

3 Model

Transformers are particularly important because they revolutionized NLP by providing a more efficient way to process sequences compared to previous recurrent-based models. They excel at handling long-range dependencies, effectively understanding and modeling relationships between elements across entire sequences. This is achieved through the attention mechanism, which dynamically adjusts the importance of different elements based on their relevance to each other. Additionally, Transformers support parallelization during training, significantly enhancing both the performance and speed of training large models. These capabilities make Transformers a powerful and versatile tool for a wide range of applications beyond NLP, including genomics, where understanding complex dependencies within sequences is crucial.

In the field of genomics, the parallels between language and genetic sequences make the implementation of Transformers particularly appealing. The attention mechanism of Transformers can effectively model dependencies between different genomic regions, capturing the interactions that define linkage disequilibrium. An example of this occurrence is shown in Figure 2, where the linkage disequilibrium is shown for a soybean protein genome.

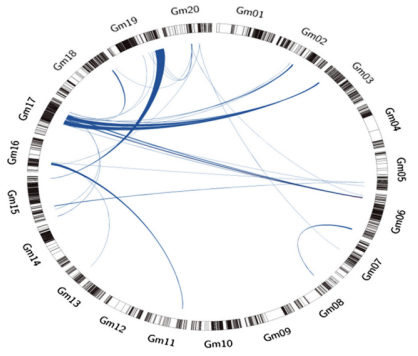

Fig. 2. Stable SNP interactions related to soybean protein content under multiple environments. The soybean genome is represented by a circle. The blue lines indicate the interactions between two markers or regions, presented by Chen et al. [2].

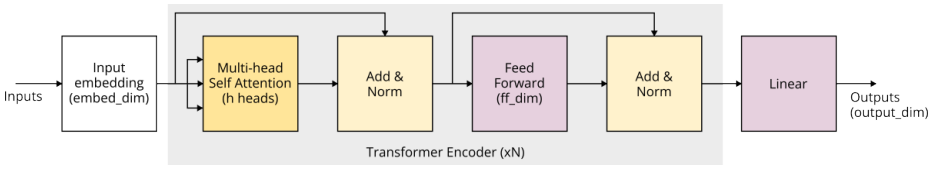

Fig. 3. Model trained with the Yeast database for predicting growth in different environments.

To predict a phenotype from the genotype, the model must learn the dependencies and semantics of the input data. In the *Transformer* algorithm, this task is performed by the *Encoder*, so the model used for this problem will not be a bidirectional *Encoder-Decoder* but will consist solely of the former.

The implemented model initially presents a linear layer functioning as an *Embedding* layer. It has as input dimension the number of SNPs (`p`) per individual and as output the hyperparameter of the dimension of the embedding space (`embed_dim`). Each of the positions that make up the individual's genotype is represented by a vector of dimension `embed_dim`, so when entering the *Encoder*, each individual is represented by a matrix of dimension $\texttt{embed\_dim} \times \texttt{p}$.

An explicit *Positional Encoding* module is not used since each position has a distinct representation at the output of the linear layer, thus preserving the positional information. The number of *Encoders* in the model is defined by the hyperparameter `NLayers`. The structure, in this case, is the same as the *Encoder* structure presented for the *Transformer*: a *Multi-Head Self Attention* block formed by `h` heads, a *Feed-Forward Neural Network* (FFN) of dimension `ff_dim`, and two *Add & Norm* layers at the output of each of the previously mentioned modules. Both `h` and `ff_dim` are hyperparameters of the model.

Finally, the model has a linear layer responsible for predicting the phenotype for each individual. The output dimension (`output_dim`) will be defined according to the number of phenotypes to be predicted with the same model ($\texttt{output\_dim} = 1$ for predicting one phenotype and $\texttt{output\_dim} = 2$ for predicting two). Figure 3 shows the diagram of the implemented model.

To implement the model, we use modules from the `Pytorch` library. In particular, the `Encoder` class *TransformerEncoder* and `nn.Linear` module for the *Embeddings* layer and output FFN.

4 Hyperparameter Search and Training

The model training was divided into two stages: first, a search for optimal hyperparameters was conducted, followed by the training of the model. All experiments were carried out on the ClusterUY [11] using a 40 GB GPU.

For the hyperparameter search, possible values for the *learning rate*, `h`, `ff_dim`, `embed_dim`, and `dropout` (used for regularization) were defined, where each training session used a different combination of these values. Finally, the combination that yielded the best value for the *Pearson Correlation Coefficient* (PCC),

$r(\mathbf{x}, \mathbf{y})$, was selected. The result is a coefficient that measures the linear dependence between variables $\mathbf{x}$ and $\mathbf{y}$, with values ranging from $[-1, +1]$. The closer r is to the extremes of the interval, the greater the linear dependence between the variables, while the closer it is to the middle 0, the lesser the dependence. Its mathematical expression is as follows:

$$r(\mathbf{x}, \mathbf{y}) = \frac{\text{cov}(\mathbf{x},\mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} = \frac{\sum_{i=0}^{n-1}(x_i - m_{\mathbf{x}})(y_i - m_{\mathbf{y}})}{\sqrt{\sum_{i=0}^{n-1}(x_i - m_{\mathbf{x}})^2 \sum_{i=0}^{n-1}(y_i - m_{\mathbf{y}})^2}}, \tag{1}$$

where $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{y}}$ are the variances of $\mathbf{x}$ and $\mathbf{y}$, respectively, and $m_{\mathbf{x}}$ and $m_{\mathbf{y}}$ are the means of $\mathbf{x}$ and $\mathbf{y}$, respectively. Initially, tests were conducted to consider the number of *Encoders* as a hyperparameter, but due to computational limitations, it was decided to train the model with two. The implementation was done using the Optuna library [1].

The available dataset was divided into five folds to enable 5-fold cross-validation training with subsequent validation. Batches of eight individuals were taken, each with a sequence of 11,623 SNPs in length.

The *Mean Squared Error* (MSE) was used as the loss function, a measure of how accurate the machine learning model is in terms of predicting the values $\tilde{y}_i = g(\mathbf{x}_i)$. It is used in cases where high sensitivity to outlier values is desired due to being squared. Its mathematical expression is:

$$\text{MSE}(\mathbf{x}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \tilde{y})^2. \tag{2}$$

The PCC (Equation (1)) was used as the gain function. This meant that for each epoch, although the parameters were updated considering the MSE, once validation was done, the best model was chosen according to the epoch with the highest PCC. It is also noteworthy that the training MSE is computed during the forward propagation of the network and the validation MSE after backpropagation, resulting in better validation results than in training. The optimizer used was *Adam* [9]. The model parameters were initialized using the *Xavier Uniform* method [6].

Training was conducted for a maximum of one thousand epochs. *Early stopping* was used as a regularization method, stopping the training if no improvements in the validation PCC were observed after thirty-five consecutive epochs. Additionally, *Dropout* [12] stages were used both in the neural network within the *Encoder* and at its output, both with the same *dropout ratio*.

5 Results

Figure 4 shows the evolution of PCC and MSE for training and validation as a function of epochs for Lactate. It can be observed that as the epochs progress, both results improve, with PCC increasing and MSE decreasing, as expected. As in the case of the simulated data, the validation results are better than those

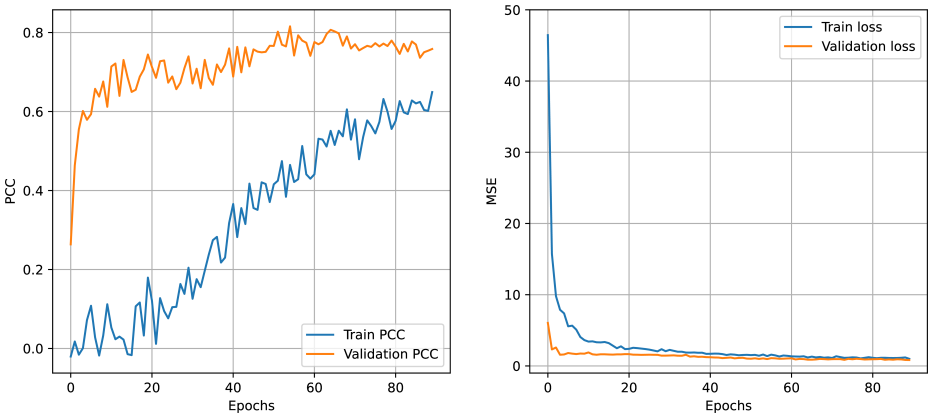

Fig. 4. Learning curves for data of yeast growth in Lactate environment. On the left is the PCC curve (used as gain function), and on the right is the MSE curve (used as the loss function). Training curves are in blue and validation curves are in orange.

for training. However, the training was stopped due to the *early stopping*. Additionally, the training PCC could have achieved a higher value if the validation PCC had not remained unchanged for approximately 30 epochs, which was the stopping condition.

In the case of Lactose, the behavior of MSE and PCC is similar to that of Lactate. It can be seen that there is some bias in the MSE and the presence of overfitting since around epoch 150. The MSE for validation surpasses that of training, as shown in Figure 5. Again, it is observed that, despite no improvement being presented for more than approximately 50 epochs in validation PCC (causing the training to stop), the training PCC could have reached a higher value, as it shows an increasing trend up to this epoch. The same could have been manifested in the training MSE with respect to the decrease.

In Figure 6, the PCC and MSE curves for each phenotype with the *multi-trait* model are presented. The results are similar to those obtained when training the model with a single phenotype: as the PCC increases, the MSE decreases. However, the MSE presents a smaller bias, and unlike in the case of Lactate, the model does not overfit.

In Figure 7, the test results are presented. In both cases, the results obtained with *multi-trait* were better than training the model with a single phenotype. Although MSE does not show significant changes, the PCC increased moderately. The relationship is not strongly linear in all cases, as the PCC is not as high as in the simulation cases.

To compare the results obtained with other models that have been trained for this dataset, the R^2 metric (*coefficient of determination*) is calculated. This metric is used to evaluate how well a model has performed on a dataset, with its best result being one and decreasing towards zero as the model's performance declines. A R^2 value of zero indicates that the model's predictions are as good as

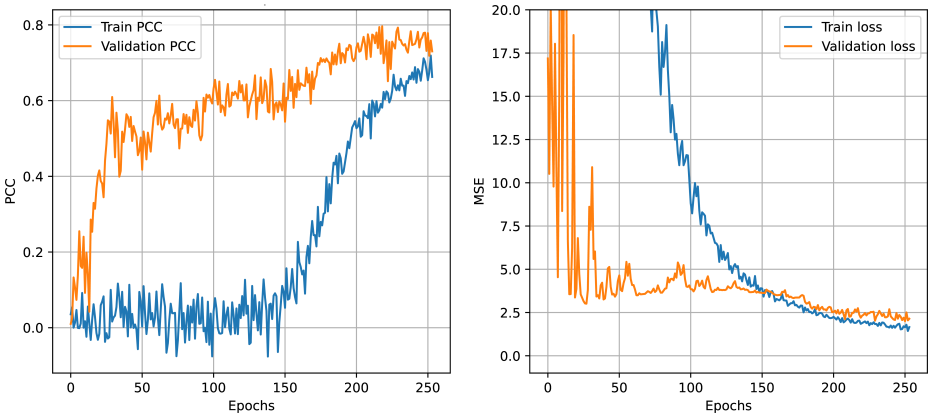

Fig. 5. Learning curves with Lactose growth data. On the left is the PCC curve (used as gain function), and on the right is the MSE curve (used as the loss function). Training curves are in blue and validation curves are in orange.

Environment	Grinberg	GBM	One trait	Multitrait
Lactate	0.568	0.830	0.389	0.478
Lactose	0.582	0.860	0.400	0.536

Table 1. Comparation of R^2 metric results for yeast growth in Lactate and Lactose obtained with both our models, Transformer One-Trait and Transformer Multitrait, with the ones reported by Grinberg et al. [7], Elenter et al. [4] .

those of a random model, while if the result is outside the interval $[0, 1]$, the model has performed worse than random predictions. The R^2 metric results, obtained for the phenotype predictions of Lactate and Lactose *One trait* and *Multitrait*, are presented in Table 1, along with other results reported by Elenter et al. [4], compared for the same phenotypes predicted by other models. It can be observed from the table that the results for the *Multi-trait* models significantly outperform the *Single-trait* models, as previously indicated. On the other hand, while these results do not reach those of Gradient Boosting Machine (GBM) [4] presented in the table, they do achieve the order of magnitude of those of Grinberg et al. [7], confirming the robustness of the implemented algorithm.

6 Conclusions

In this paper, we have experimented with Transformers applied to genomic prediction. We have described the different considerations taken to do the hyperparameter tunning and train the model.

The results obtained are promising and allow us to affirm that it is feasible to achieve satisfactory results by using models based on *Self-Attention* on ge-

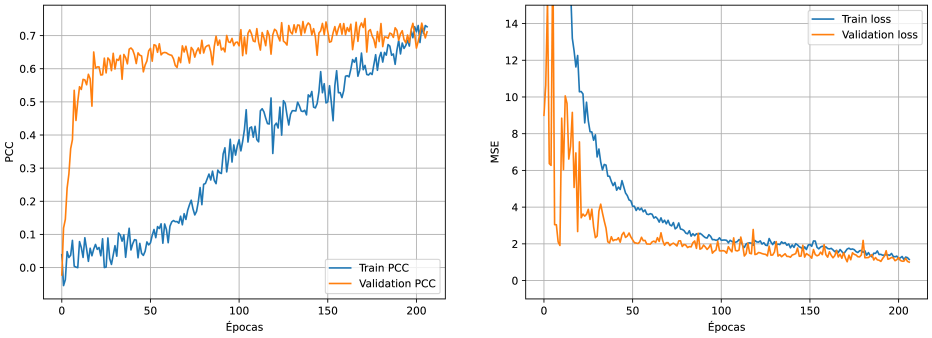

Fig. 6. Learning curves for yeast growth in Lactate and Lactose, predicted using the Multitrait model. On the left is the PCC curve which shows the average of the PCC obtained for the two phenotypes in training (blue) and validation (orange). On the right, the MSE curves, with training shown in blue and validation in orange.

nomic data sequences. However, there are modifications, validations, and new simulations that need to be explored, including:

1. More exhaustive hyperparameter searches.
2. Balancing choosing the best models according to PCC and MSE.
3. Investigating better parameter initialization methods.
4. Repeating the process on new datasets.

The first point is directly related to the computing power available. The maximum GPU memory accessible for us was 40 GB, and it was with this that hyperparameter searches were performed using *Optuna*. For all searches conducted, the best parameters obtained were always the maximum of the intervals studied, indicating that larger hyperparameter intervals must be studied. This could not be done as memory saturation was reached in all cases.

The second point on the list is due to the obtained results for Lactate, where overfitting is observed in the training curve with the MSE metric. This is related to the fact that the stopping condition and the choice of the best epoch were made based on the PCC coefficient, but a way should have been found to balance, and include, the results for each epoch of the MSE metric. Although the best epoch was tried according to the MSE metric, and the results were not better, other alternatives could have been considered, such as combining both PCC and MSE results.

In conclusion, although this work has shown positive results and promises great potential, the implementation of additional modifications and validations, as well as the exploration of new simulations and methods, are necessary to continue improving the accuracy and robustness of the model. Exploring these aspects will establish a solid foundation for future research and applications in the field of genomics using models based on *Transformers*.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
2. Chen, Q., Qi, H., Zhang, X., Li, W., Hou, M., Zhu, R., Yin, Z., Han, X., Jiang, H., Liu, C., et al.: Snp–snp interaction analysis of soybean protein content under multiple environments. Canadian Journal of Plant Science **97**(6), 1090–1099 (2017)
3. Clauwaert, J., Menschaert, G., Waegeman, W.: Explainability in transformer models for functional genomics. Briefings in bioinformatics **22**(5), 1–11 (2021)
4. Elenter, J., Etchebarne, G., Hounie, I.: DNAI: Machine learning for genome enabled prediction of complex traits in agriculture. Master's thesis (2021)
5. Gill, H.S., Halder, J., Zhang, J., Brar, N.K., Rai, T.S., Hall, C., Bernardo, A., Amand, P.S., Bai, G., Olson, E., et al.: Multi-trait multi-environment genomic prediction of agronomic traits in advanced breeding lines of winter wheat. Frontiers in Plant Science **12**, 709545 (2021)
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy (13–15 May 2010), `https://proceedings.mlr.press/v9/glorot10a.html`
7. Grinberg, N.F., Orhobor, O.I., King, R.D.: An evaluation of machine-learning for predicting phenotype: studies in yeast, rice, and wheat. Machine Learning **109**, 251–277 (2020)
8. Jubair, S., Tucker, J.R., Henderson, N., Hiebert, C.W., Badea, A., Domaratzki, M., Fernando, W.: Gptransformer: A transformer-based deep learning method for predicting fusarium related traits in barley. Frontiers in plant science **12**, 2984 (2021)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Moeinizade, S., Kusmec, A., Hu, G., Wang, L., Schnable, P.S.: Multi-trait genomic selection methods for crop improvement. Genetics **215**(4), 931–945 (2020)
11. Nesmachnow, S., Iturriaga, S.: Cluster-uy: Collaborative scientific high performance computing in uruguay. In: Torres, M., Klapp, J. (eds.) Supercomputing. pp. 188–202. Springer International Publishing, Cham (2019)
12. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

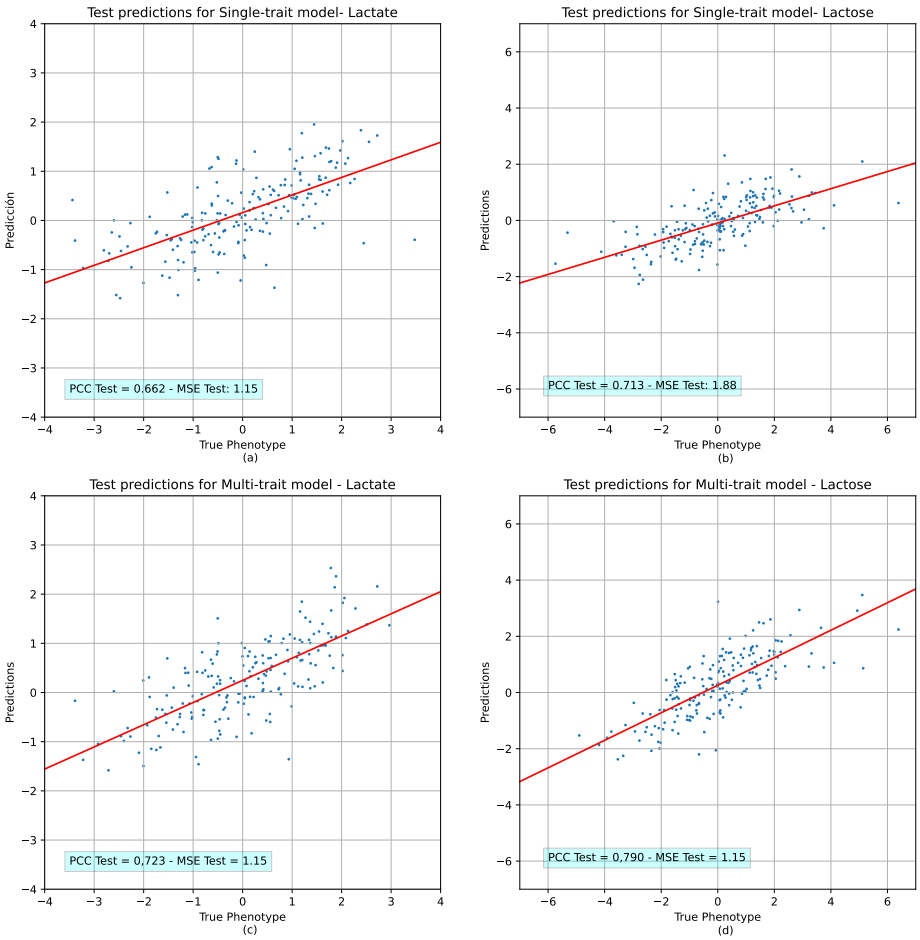

Fig. 7. Scatter plot of true phenotype values of the test set versus their predictions. In figure (a) the results of One-trait model used to predict phenotype in Lactate, in figure (b) One-trait model results for Lactose. Figures (c) and (d) show the results for the Multitrait model in Lactate and Lactose environments respectively. In all figures, the linear fit is shown in red.