# Supplementary Information

# Performance of regression models as a function of experiment noise

Gang Li[1#], Jan Zrimec[1#], Boyang Ji[1,2], Jun Geng[1], Johan Larsbrink[1], Aleksej Zelezniak[1,3], Jens Nielsen[1,2,4], and Martin KM Engqvist[1*]

[1] Department of Biology and Biological Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

[2] Novo Nordisk Foundation Center for Biosustainability, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

[3] Science for Life Laboratory, Tomtebodavägen 23a, SE-171 65, Stockholm, Sweden

[4] BioInnovation Institute, Ole Måløes Vej 3, DK-2200 Copenhagen N, Denmark

* Corresponding author

# These authors contributed equally.

E-mail: martin.engqvist@chalmers.se

## Table of contents

# Supplementary Notes

**Note S1: The expectation and variance of best $R^2$score**

Given a set of samples with experimentally determined labels $\{y_{obs,i}\}$ and corresponding unknown real labels $\{y_i\}$, By assuming a normally distributed experimental noise term$\varepsilon_{y,i} \sim N(0, \sigma_{y,i}), y_{obs,i} = y_i + \varepsilon_{y,i}$ $(y_i \in R)$. A complete set of features is known as $x_i \in R^k$ for each sample. The "complete" means that this set of features are sufficient to accurately calculate the real value of label $y_i$ with $y = f(x)$ for all samples. The performance of this real function $f(x)$ on the dataset $\{x_i, y_{obs,i}\}$ gives an upper bound for the expected performance of any ML model. The coefficient of determination $(R^2)$ of the model $f(x)$ in the above argument is given by

$$R^2 = 1 - \frac{\sum_{i=1}^{m} (y_{obs,i} - \hat{y}_{obs,i})^2}{\sum_{i=1}^{m} (y_{obs,i} - \underline{y}_{obs})^2} = 1 - \frac{\sum_{i=1}^{m} (y_{obs,i} - f(x_i))^2}{\sum_{i=1}^{m} (y_{obs,i} - \underline{y}_{obs})^2}$$

where $m$ is the number of samples.

$$R^2 = 1 - \frac{\sum_{i=1}^{m} (y_{obs,i} - \hat{y}_{obs,i})^2}{\sum_{i=1}^{m} (y_{obs,i} - \underline{y}_{obs})^2}$$

Since $f(x_i) = y_i$, $y_{obs,i} - f(x_i) = y_{obs,i} - y_i = \varepsilon_{y,i}$, thereby the numerator is $\sum_{i=1}^{m} (y_{obs,i} - f(x_i))^2 = \sum_{i=1}^{m} \epsilon_{y,i}^2$. The expectation is given by

$$\langle R^2 \rangle = 1 - \langle \frac{\sum_{i=1}^{m} \epsilon_{y,i}^2}{\sum_{i=1}^{m} (y_{obs,i} - \underline{y}_{obs})^2} \rangle = 1 - \sum_{i=1}^{m} \langle \frac{\epsilon_{y,i}^2}{\sum_{j=1}^{m} (y_{obs,i} - \underline{y}_{obs})^2} \rangle.$$

Since $\epsilon_{y,i}$is normally distributed with a zero-mean and variance of $\sigma_{y,i}^2$, then $\frac{\varepsilon_{y,i}}{\sigma_{y,i}}$ follows a standard normal distribution. Thereby $(\frac{\varepsilon_{y,i}}{\sigma_{y,i}})^2$follows a chi-squared distribution with a degree of 1 $(\chi^2(1))$. The numerator becomes $\epsilon_{y,i}^2 = \sigma_{y,i}^2 \frac{\epsilon_{y,i}^2}{\sigma_{y,i}^2} \sim \sigma_{y,i}^2 \cdot \chi^2(1)$. We assume that the variance of the observed values $y_{obs,i}$ is normally distributed with a variance of $\sigma_{obs}^2$, then

$$\sum_{j=1}^{m} (y_{obs,i} - \underline{y}_{obs})^2 \sim \sigma_{obs}^2 \cdot \chi^2(m-1).$$

The ratio between two chi-squared distributions is an $F$ distribution multiplied by the ratio between their degrees of freedom, thereby

$$\langle R^2 \rangle = 1 - \sum_{i=1}^{m} \frac{\sigma_{y,i}^2}{\sigma_{obs}^2} \langle \frac{\chi^2(1)}{\chi^2(m-1)} \rangle = 1 - \sum_{i=1}^{m} \frac{\sigma_{y,i}^2}{\sigma_{obs}^2} \frac{1}{m-1} \langle F(1, m-1) \rangle.$$

Since $\langle F(1, m-1) \rangle = \frac{m-1}{m-3}$, then

$$\langle R^2 \rangle = 1 - \frac{1}{m-3} \sum_{i=1}^{m} \frac{\sigma_{y,i}^2}{\sigma_{obs}^2} = 1 - \frac{m}{m-3} \frac{\overline{\sigma_y^2}}{\sigma_{obs}^2}$$

The variance is given by

$$Var(R^2) = Var(1 - \frac{\sum_{i=1}^{m} (y_{obs,i} - f(x_i))^2}{\sum_{i=1}^{m} (y_{obs,i} - \underline{y}_{obs})^2}) = Var(\frac{\sum_{i=1}^{m} (y_{obs,i} - f(x_i))^2}{\sum_{i=1}^{m} (y_{obs,i} - \underline{y}_{obs})^2})$$

With a similar approach as for expectation,

$$Var(R^2) = Var(\sum_{i=1}^{m} \frac{\sigma_{y,i}^2}{\sigma_{obs}^2} \frac{1}{m-1} \frac{\epsilon_{y,i}^2/\sigma_{y,i}^2}{(\sum_{i=1}^{m} (y_{obs,i} - \underline{y}_{obs})^2/\sigma_{obs}^2)/(m-1)})$$

$\frac{\sigma_{y,i}^2}{\sigma_{obs}^2}$ is a constant and the expectation of the ratio part is $\frac{2(m-1)^2(m-2)}{(m-3)^2(m-5)}$, thereby

$$Var(R^2) = \frac{1}{(m-1)^2} \frac{2(m-1)^2(m-2)}{(m-3)^2(m-5)} \sum_{i=1}^{m} \frac{\sigma_{y,i}^4}{\sigma_{obs}^4} = \frac{2m(m-2)}{(m-3)^2(m-5)} \frac{\underline{\sigma_y^4}}{\sigma_{obs}^4}$$

**Note S2: The expectation and variance of MSE**

The expectation of MSE on the dataset $\{x_i, y_{obs,i}\}$is given by

$$\langle MSE \rangle = \langle \frac{1}{m} \sum_{i=1}^{m} (y_{obs,i} - f(x_i))^2 \rangle$$

Since $f(x_i) = y_i$, $y_{obs,i} - f(x_i) = y_{obs,i} - y_i = \varepsilon_{y,i}$, thereby

$$\langle MSE \rangle = \frac{1}{m} \langle \sum_{i=1}^{m} \epsilon_{y,i}^2 \rangle = \frac{1}{m} \sum_{i=1}^{m} \sigma_{y,i}^2 \langle \frac{\epsilon_{y,i}^2}{\sigma_{y,i}^2} \rangle$$

Since $\epsilon_{y,i}$is normally distributed with a zero mean and variance of $\sigma_{y,i}^2$, then $\frac{\varepsilon_{y,i}}{\sigma_{y,i}}$ follows a standard normal distribution. Thereby $(\frac{\varepsilon_{y,i}}{\sigma_{y,i}})^2$follows a chi-squared distribution with a degree of 1 ($\chi^2(1)$). The expectation of this $\chi^2(1)$ is 1, thereby

$$\langle MSE \rangle = \frac{1}{m} \sum_{i=1}^{m} \sigma_{y,i}^2 = \underline{\sigma_y^2}$$

This gives a lower bound of expected MSE values for machine learning models.

Accordingly the variance of MSE is given by

$$Var(MSE) = Var(\frac{1}{m} \sum_{i=1}^{m} (y_{obs,i} - f(x_i))^2) = Var(\frac{1}{m} \sum_{i=1}^{m} \sigma_{y,i}^2 \frac{\epsilon_{y,i}^2}{\sigma_{y,i}^2})$$

The $\sigma_{y,i}^2$ is a constant and the variance of $\frac{\epsilon_{y,i}^2}{\sigma_{y,i}^2} \sim \chi^2(1)$ is 2. Thereby

$$Var(MSE) = \frac{2}{m^2}\sum_{i=1}^{m} \quad \sigma_{y,i}^4 = \frac{2\sigma_y^4}{m}$$

# Supplementary Figures

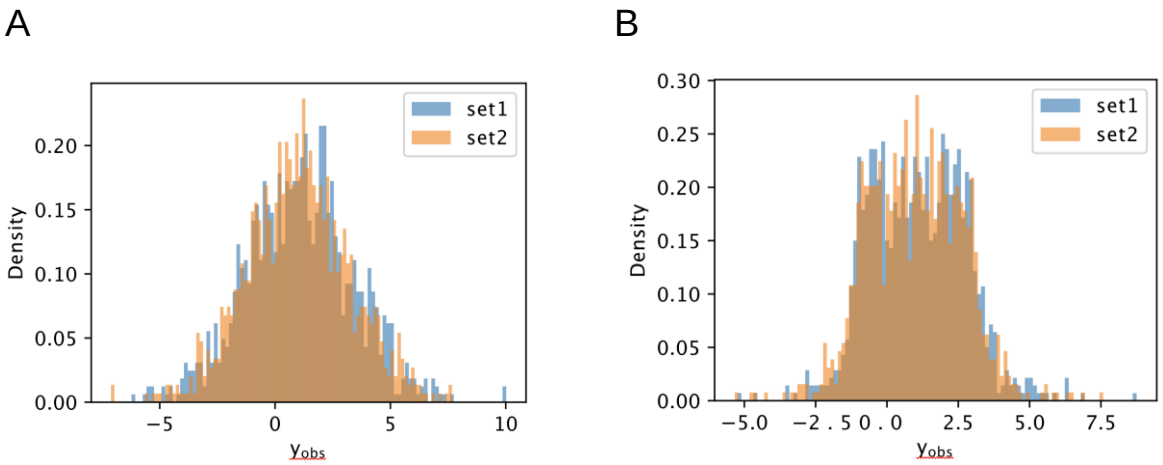


**Figure S1.** Examples of the data distributions of two sets of $y_{obs}$ each for the (A) linear and (B) nonlinear functions used for Monte Carlo simulations of the upper bound of $R^2$ assuming different levels of feature noise (corresponding to Figures 2A and 2B, respectively).

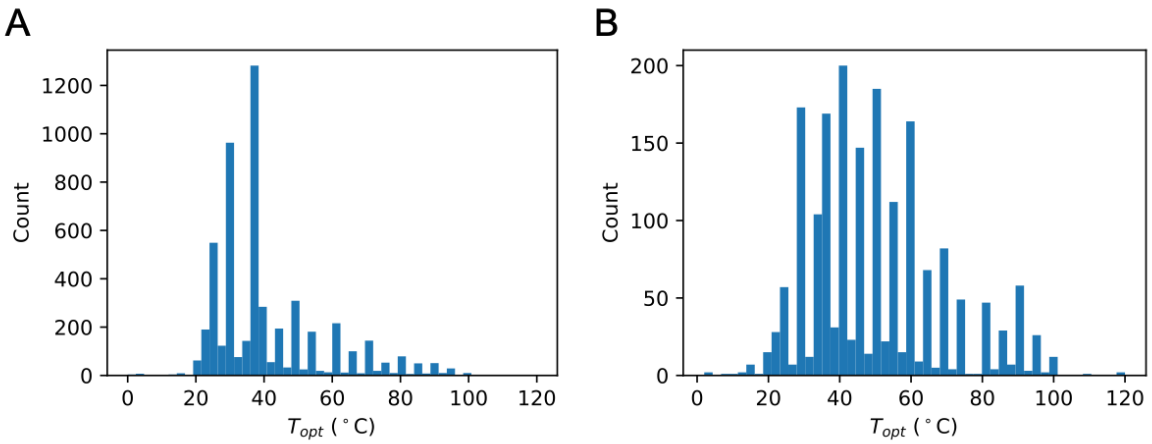


**Figure S2.** Distribution of enzyme $T_{opt}$ values in the dataset (A) before and (B) after cleaning.

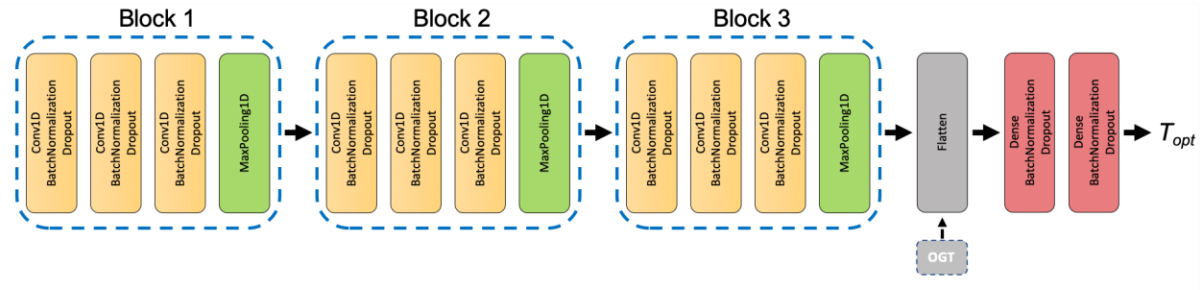


**Figure S3.** Deep NN architecture. There are three convolution layers in each of three blocks and have the same hyper-parameters. The hyper-parameter space for optimization with Hyperopt[1] is listed in Table S3.

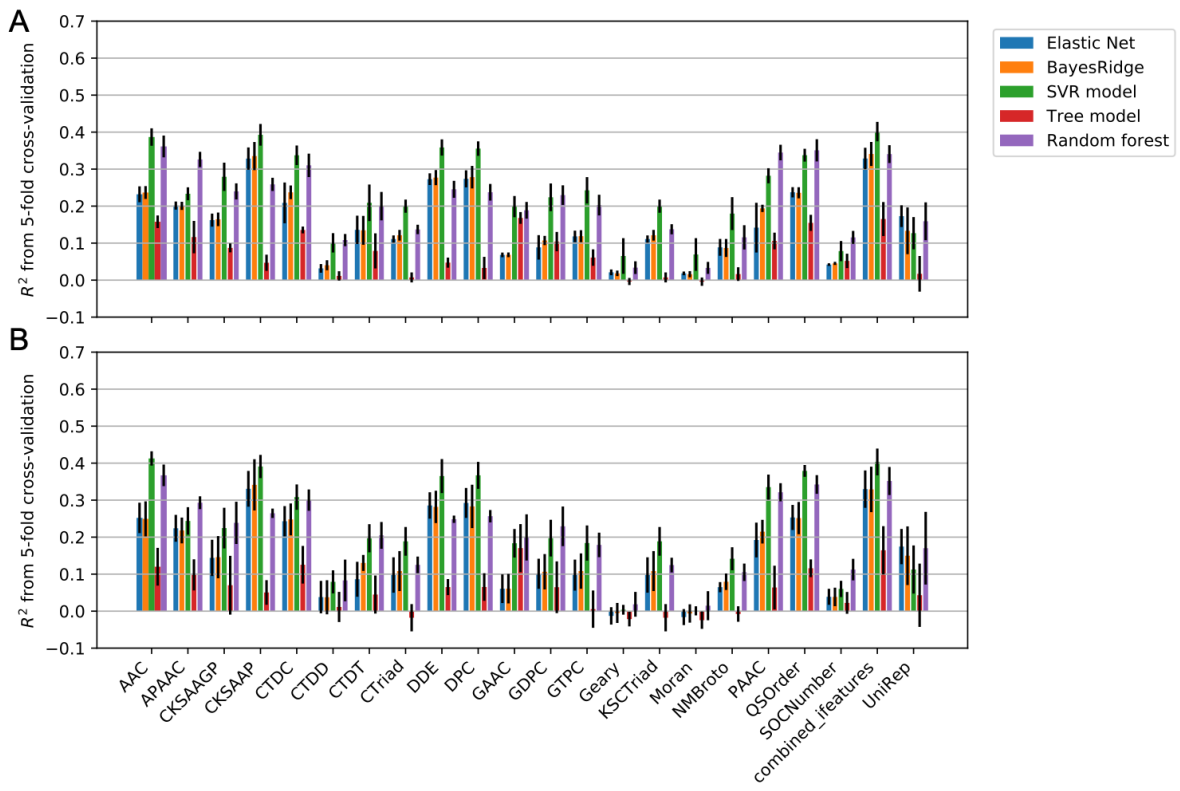


**Figure S4.** Models trained using amino acid composition (20 features) showed the same predictive performance as the whole iFeature set, both with and without OGT as an extra feature, as well as before and after data cleaning. This showed that, compared to the amino acid composition, the 5,454 additional features derived from the protein sequence did not carry additional information for predicting enzyme $T_{opt}$. Future improvement of $T_{opt}$ prediction therefore necessitates that more relevant features are engineered, for instance ones extracted from protein 3D structures. The plots show the performance of five regression models when trained on different feature sets **without OGT** as an additional feature, with (A) the dataset before cleaning and (B) the dataset after data cleaning. Detailed description of those feature sets can be found in **Methods 4.6**. Error bars show the standard deviation of $R^2$ scores obtained in 5-fold cross validation.

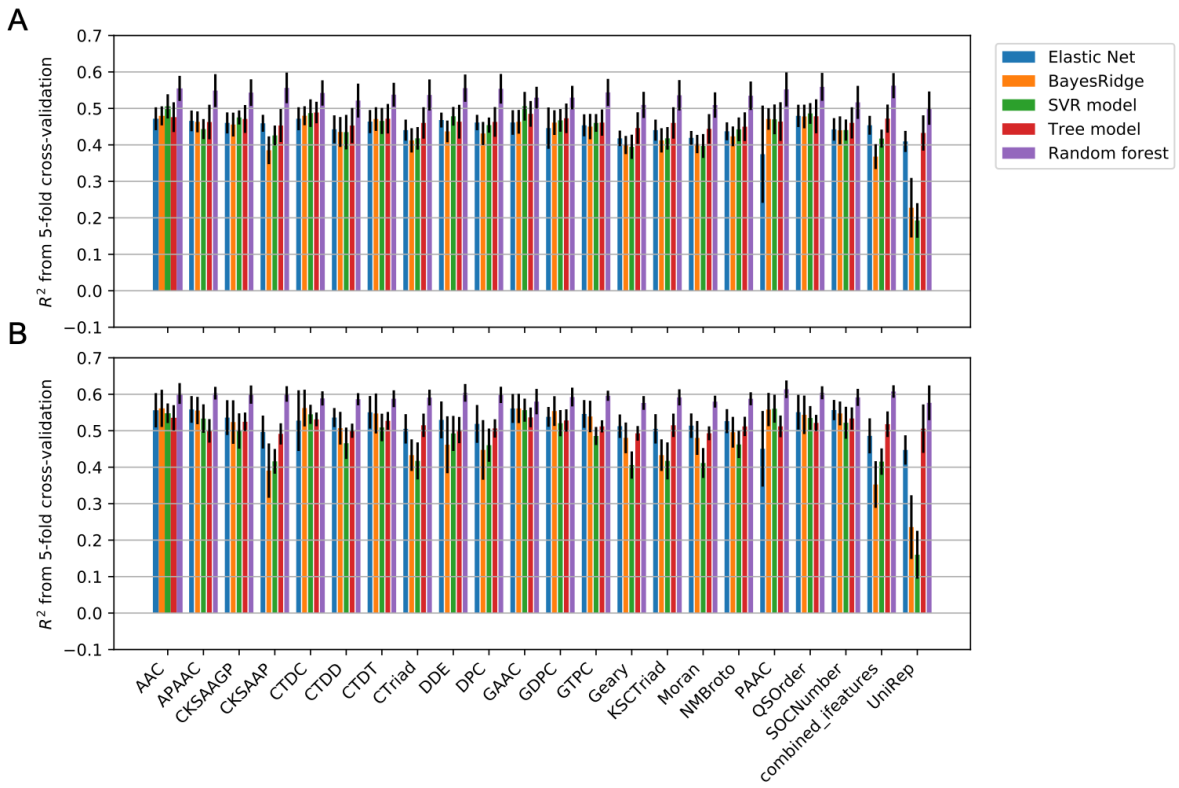


**Figure S5.** The performance five regression models when trained on different feature sets **with OGT** as an additional feature, with (A) the dataset before cleaning and (B) the dataset after data cleaning. Detailed description of those feature sets can be found in **Methods 4.6.** Error bars show the standard deviation of $R^2$ scores obtained in 5-fold cross validation.

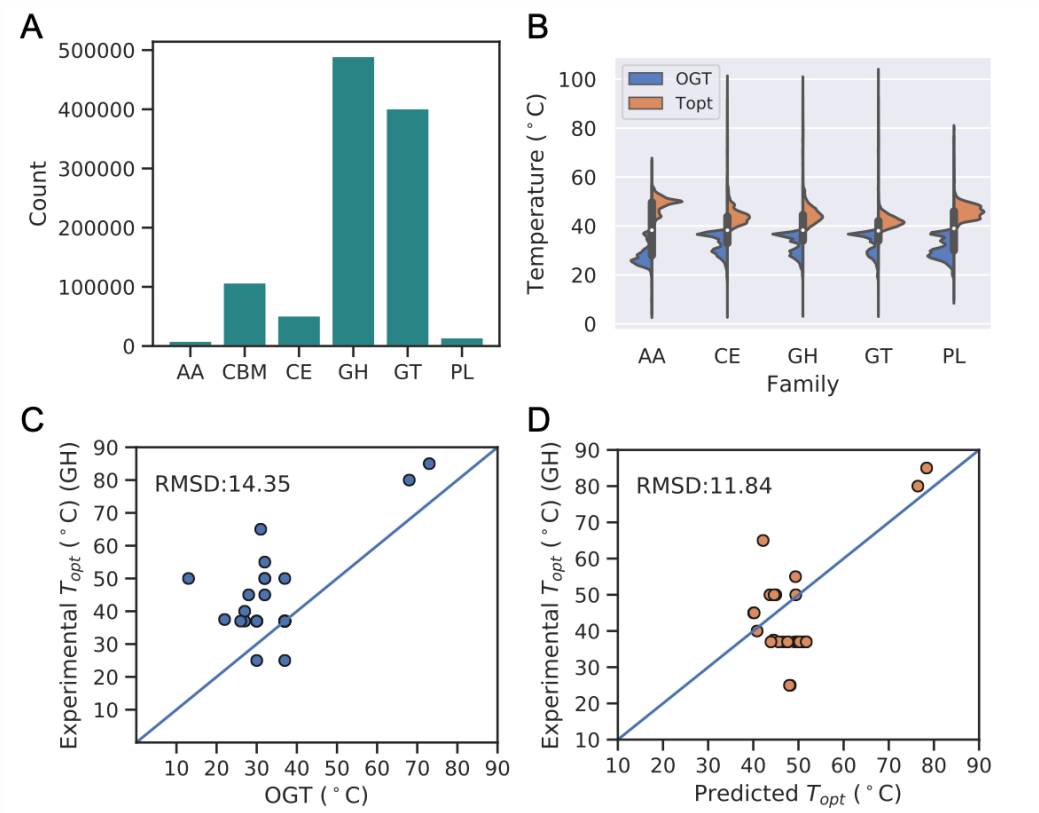


**Figure S6.** Predict $T_{opt}$ of CAZy enzymes [2]. (A) 924, 642 sequences covering 6 CAZy families can be mapped to an optimal growth temperature (OGT) value by cross-referencing the source organism name and an OGT dataset[3]. The distribution of OGT and predicted $T_{opt}$ of each CAZy family was shown in (B). A list of commercialized enzyme $T_{opt}$ values from nzytech (https://www.nzytech.com/) were collected to validate our predictions. nzytech data were downloaded from https://www.nzytech.com/resources/catalogues/. A pdf file cazymes_2019.pdf was downloaded. Then this pdf file was parsed to obtain the CAZy family id, source organism name and optimal temperature of all enzymes in the file. Since there is no sequence provided, nor any sequence/gene id that could be mapped to a sequence database, it's impossible to exactly map those enzymes to the ones in CAZy database. Thereby we used the following strategies to do the mapping: for a given CAZy family id from a specific organism, if there is only one record in nzytech dataset and also only one record in CAZy dataset, then we consider those two enzymes are the same enzyme. In such a way, we could find experimental $T_{opt}$ values from nzytech dataset. To validate our prediction, the enzymes in the training dataset were also removed by comparing protein sequences of those CZAy enzymes to ones in the training dataset. In the end, 27 enzymes from family GH were obtained (there are only less than 10 enzymes were found for other families, then they are not included in comparison). Even though our prediction is still not a perfect estimation of experimental values ( RMSE: 11.84 °C ) , this is a more accurate estimation than OGT values (Figure S3C and S3D). AA: Auxiliary Activity, CBM: Carbohydrate-Binding Module, CE: Carbohydrate Esterase, GH: Glycoside Hydrolase, GT: Glycosyl Transferase, PL: Polysaccharide Lyase.

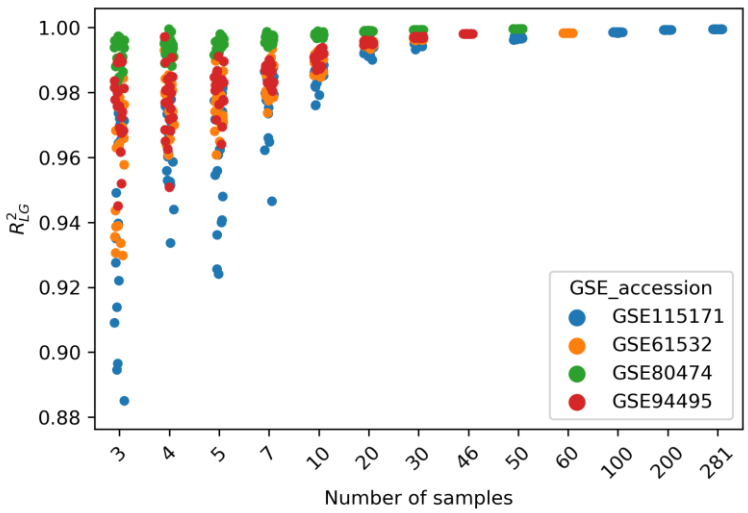


**Figure S7.** The estimated upper bounds for condition-specific subsets of the transcriptomics dataset. NCBI Gene Expression Omnibus (https://www.ncbi.nlm.nih.gov/geo/) identifiers GSE were used to group the data across conditions and $\langle R^2 \rangle_{LG}$ was estimated for the four largest subsets with 281, 60, 50 and 46 samples, respectively.

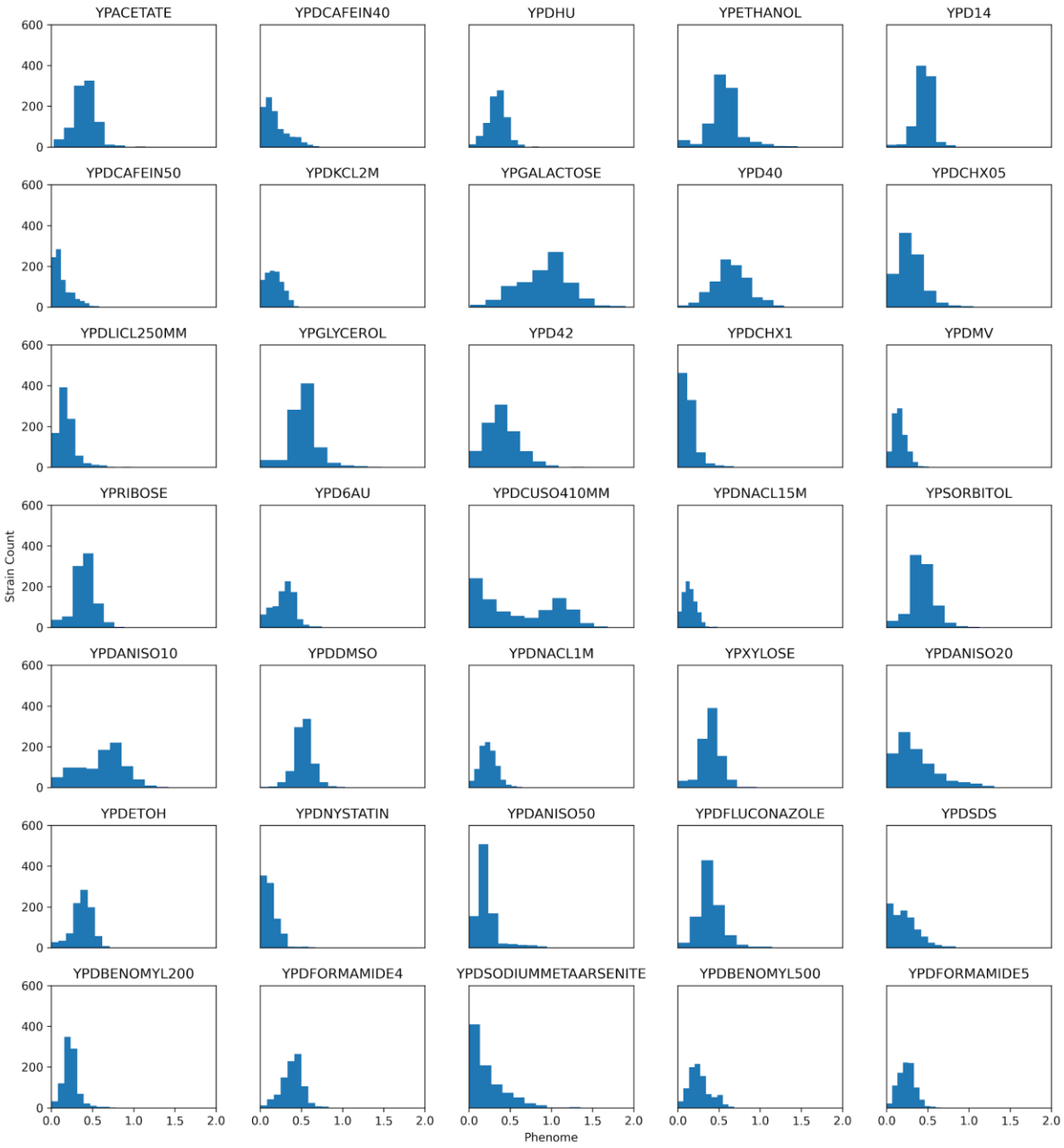


**Figure S8**. Distribution of 35 quantitative traits collected from Peter J et al [4].

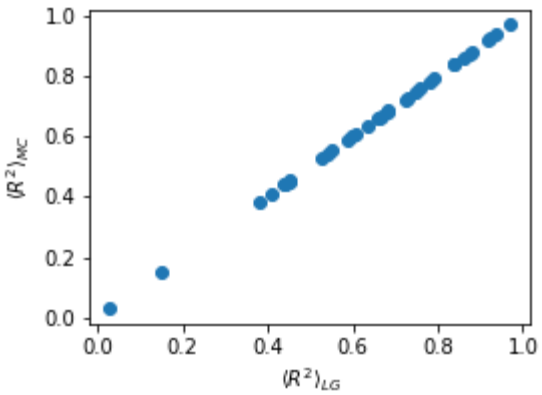


**Figure S9.** Comparison between $\langle R^2 \rangle_{LG}$and $\langle R^2 \rangle_{MC}$for datasets of 35 quantitative traits collected from Peter J et al [4].

# Supplementary Tables

**Table S1.** The estimated $\langle R^2 \rangle_{LG}$ for melting temperature datasets from Leuenberger *et al* [5].

| | $\sigma_y^2$ | $\sigma_{obs}^2$ | $\langle R^2 \rangle_{LG}$ |
|---|---|---|---|
| *S. cerevisiae* | $1.56^2$ | $5.89^2$ | 0.93 |
| *E. coli* | $1.31^2$ | $7.39^2$ | 0.97 |
| Human Hela cell | $5.49^2$ | $6.57^2$ | 0.30 |
| *T. thermophilus* | $1.29^2$ | $8.02^2$ | 0.97 |

**Note:** Leuenberger R et al [5] measured melting temperatures ($T_m$ values) of 3,557 proteins from *Escherichia coli* (730), *Saccharomyces cerevisiae* (707), *Thermus thermophilus* (1,083), and human Hela cells (1,037) via a proteomics approach. In this approach, proteins were first digested into peptides by limited proteolysis. Then $T_m$s of those peptides were measured. Thirdly, peptides with high-quality $T_m$ values were clustered the average $T_m$ were assigned as the $T_m$ of this cluster. At last, the cluster with the lowest $T_m$ was assigned as the $T_m$ of the protein. Since the standard error was not reported for protein $T_m$ values, the reported 95% confidence interval of single peptides were used to estimate the standard error of protein $T_m$ values with following approach: 1) calculate the standard error of each peptide listed Table S3 of [5] from its 95% confidence interval listed in Table S3 of [5] as (tm_ciu-tm_cil)/2/1.96, in which tm_ciu and tm_cil are the upper and lower bounds; 2) for a dataset with a list of proteins from considered organism(s), calculate the average squared standard errors of the peptides in the dataset; 3) estimate the average number of peptides in each protein in the considered dataset by dividing the number of peptides in each protein by the theoretical number of domains (from Table S3 of [5]); 4) the average peptide standard error from step 2) was divided by the root of the average peptide number obtained from step 3). This value was considered as an approximation of the average standard error $\sqrt{\sigma_y^2}$ of the considered dataset.

**Table S2.** Regression models used and the corresponding hyper-parameter spaces.

| **Regression model** | **Module** | **Hyperparameter range** |
|---|---|---|
| Linear model | sklearn.linear_model.LinearRegression | None |
| Elastic net | sklearn.linear_model.ElasticNetCV | Default |
| Bayes ridge | sklearn.linear_model.BayesianRidge | None |
| Support vector regressor | sklearn.svm.SVR | 'C': numpy.logspace(-5, 10, num=16, base=2.0), 'Epsilon': [0, 0.01, 0.1, 0.5, 1.0, 2.0, 4.0] |
| Decision tree | sklearn.tree.DecisionTreeRegressor | 'Min_samples_leaf': numpy.linspace(0.01, 0.5, 10) |
| Random forest | sklearn.ensemble.RandomForestRegressor | 'Max_features': numpy.arange(0.1, 1.1, 0.1) |

**Table S3.** The investigated hyper-parameter space of the deep neural network (Figure S3).

| | **parameter** | **Range** |
|---|---|---|
| Block 1 | kernel size | [20, 30, 40] |
| | filter | [32, 64] |
| | stride | [2, 4, 8] |
| | dilation | [1, 2, 4] |
| | pool size | [2, 4, 8] |
| | drop out | (0, 0.4) |
| Block 2 | kernel size | [10, 20, 30] |
| | filter | [64, 128] |
| | stride | [1, 2] |
| | dilation | [1, 2, 4] |
| | pool size | [1, 2, 4] |
| | drop out | (0, 0.4) |
| Block 3 | kernel size | [10, 20] |
| | filter | [128, 256] |
| | stride | [1, 2] |
| | dilation | [1, 2, 4] |
| | pool size | [1, 2, 4] |
| | drop out | (0, 0.4) |
| 1st dense layer | size | [64, 128] |
| | drop out | (0, 0.3) |

| | | | |
|---|---|---|---|
| | | size | [32, 64] |
| 2nd dense layer | | | |
| | drop out | | (0, 0.3) |

# Supplementary References


1. Bergstra, J., Komer, B., Eliasmith, C., Yamins, D. & Cox, D. D. Hyperopt: a Python library for model selection and hyperparameter optimization. *Comput. Sci. Discov.* **8**, 014008 (2015).
2. Lombard, V., Golaconda Ramulu, H., Drula, E., Coutinho, P. M. & Henrissat, B. The carbohydrate-active enzymes database (CAZy) in 2013. *Nucleic Acids Res.* **42**, D490–5 (2014).
3. Engqvist, M. K. M. Correlating enzyme annotations with a large set of microbial growth temperatures reveals metabolic adaptations to growth at diverse temperatures. *BMC Microbiol.* **18**, 177 (2018).
4. Peter, J. *et al.* Genome evolution across 1,011 Saccharomyces cerevisiae isolates. *Nature* **556**, 339–344 (2018).
5. Leuenberger, P. *et al.* Cell-wide analysis of protein thermal unfolding reveals determinants of thermostability. *Science* **355**, (2017).