Performance of Regression Models as a Function of Experiment Noise

Citation for the original published paper (version of record):
Li, G., Zrimec, J., Ji, B. et al (2021). Performance of Regression Models as a Function of Experiment Noise. Bioinformatics and Biology Insights, 15. http://dx.doi.org/10.1177/11779322211020315

Performance of Regression Models as a Function of Experiment Noise

Bioinformatics and Biology Insights
Volume 15: 1–10
© The Author(s) 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/11779322211020315
SAGE

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, Gothenburg, Sweden. [2]Novo Nordisk Foundation Center for Biosustainability, Technical University of Denmark, Kgs. Lyngby, Denmark. [3]Science for Life Laboratory, Stockholm, Sweden. [4]BioInnovation Institute, Copenhagen N, Denmark.

ABSTRACT

BACKGROUND: A challenge in developing machine learning regression models is that it is difficult to know whether maximal performance has been reached on the test dataset, or whether further model improvement is possible. In biology, this problem is particularly pronounced as sample labels (response variables) are typically obtained through experiments and therefore have experiment noise associated with them. Such label noise puts a fundamental limit to the metrics of performance attainable by regression models on the test dataset.

RESULTS: We address this challenge by deriving an expected upper bound for the coefficient of determination (R^2) for regression models when tested on the holdout dataset. This upper bound depends only on the noise associated with the response variable in a dataset as well as its variance. The upper bound estimate was validated via Monte Carlo simulations and then used as a tool to bootstrap performance of regression models trained on biological datasets, including protein sequence data, transcriptomic data, and genomic data.

CONCLUSIONS: The new method for estimating upper bounds for model performance on test data should aid researchers in developing ML regression models that reach their maximum potential. Although we study biological datasets in this work, the new upper bound estimates will hold true for regression models from any research field or application area where response variables have associated noise.

KEYWORDS: machine learning, experiment noise, label noise, regression models, upper bound

RECEIVED: February 15, 2021. **ACCEPTED:** April 29, 2021.

TYPE: Original Research

FUNDING: The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: GL and JN have received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie program, project PAcMEN (grant agreement no. 722287). JN also acknowledges funding from the Novo Nordisk Foundation (grant no. NNF10CC1016517), the Knut and Alice Wallenberg Foundation. JZ and AZ are supported by SciLifeLab funding. The computations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC).

DECLARATION OF CONFLICTING INTERESTS: The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

CORRESPONDING AUTHOR: Martin KM Engqvist, Department of Biology and Biological Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden. Email: martin.engqvist@chalmers.se

Background

Machine learning (ML) based regression analysis, which is used to discover complex relationships between sample features and labels (supervised learning), is frequently applied in diverse biological fields including metabolic engineering,[1] protein engineering,[2] and systems biology.[3-5] A key question when developing these supervised ML models is whether there is sufficient information in the available data to accurately predict sample labels. For a given dataset, the performance of the best possible function for mapping input features to sample labels should thus be estimated to serve as a benchmark in ML model development. This level of performance is typically referred to as Bayes optimal error for classification problems.[6] In many classical ML problems—such as image classification, handwriting recognition and speech recognition—human-level performance at the task is very high and can therefore be used as a heuristic to estimate maximal performance.[7,8] However, for biological multi-dimensional data, human-level performance is low and is therefore not a good performance estimate. On the contrary, in biology one often seeks to train ML models for the explicit purpose of recognizing patterns and gaining insights that were obscured from the human intellect.[9,10] Therefore, without a clear performance benchmark against which to bootstrap biological regression models, it is difficult to know whether further model development is worth-while and when the performance limit has been reached.

When testing the performance of trained ML regression models on holdout data, the discrepancy between predicted labels and observed labels in a test dataset is evaluated using metrics such as the mean squared error (MSE) and the coefficient of determination (R^2).[11] Sample labels used in regression analysis of biological systems are typically real numbers obtained through measurements in a set of laboratory experiments. Such measurements inextricably have experimental noise and measurement error associated with them,[12-14] thus affecting the quality of the sample labels. Because of such label noise a ML model with an MSE of 0 or R^2 of 1 (perfect prediction) cannot be achieved; there is an upper bound that cannot be surpassed.

[*] Gang Li and Jan Zrimec contributed equally to this study.

Methods to estimate this upper bound are underdeveloped, although some progress has been made recently.[15,16] Moreover, the resources invested into model development have diminishing returns on model performance as one approaches the upper bound. Knowing the best expected MSE or R^2 (i.e. the upper bounds) of a specific regression problem and dataset enables the discrepancy between current and potential model performance to be quantified, thus giving researchers a means to assess the cost-benefit trade-off of further model development.

In the present study, we mathematically derive a method to estimate upper bounds for the regression model performance metrics MSE and R^2 on holdout data directly from the experimental noise associated with response variables in a dataset and independently of their predictors. Using Monte Carlo simulations, we show that this method is highly accurate and outperforms existing ones. Furthermore, by applying the method to real biological modeling problems and datasets, including protein sequence data, transcriptomics data and genomics data, we demonstrate how the new upper bound estimates can inform model development.

Methods

Enzyme catalytic temperature optima dataset

We first collected T_{opt} of 5675 enzymes with known protein sequences from BRENDA.[17] Of these 3096 enzymes were successfully mapped to a microbial optimal growth temperature (OGT) database.[18] To obtain a clean dataset with less noise we carried out several steps: (1) Enzymes for which the T_{opt} entry contained "assay at" in the BRENDA "comments" field were removed from the raw dataset. (2) If a subset of all enzymes from a specific organism had the same EC number and exactly the same T_{opt}, then these were removed. This was done to address an issue with non-perfect matching between experimental data from the literature and Uniprot identifiers (186 enzymes). (3) Enzymes with multiple T_{opt} values having standard deviations greater than 5 were removed (96 enzymes). After these steps, 1902 enzymes remained in the cleaned dataset, of which 1232 were with known OGT. In both raw and cleaned datasets, any sequences shorter than 30 residues or containing letters that are not in 20 standard amino acids were discarded and for enzymes still with multiple T_{opt} values the average value was used. Estimation of label noise: For enzymes with multiple T_{opt} values in BRENDA, the variance for each enzyme was calculated. Subsequently, the average variance for all those enzymes was calculated and used as the estimation of experimental noise $\overline{\sigma_y^2}$ of the dataset. For the cleaned dataset the label noise was estimated at step (2) in the paragraph above, before samples with high standard deviation were removed.

Protein transcription level dataset

Genomic data including open reading frame (ORF) boundaries of *Saccharomyces cerevisiae* C288 was obtained from the Saccharomyces Genome Database (https://www.yeastgenome.org/)[19,20] and published data.[21,22] Coding regions were extracted based on ORF boundaries and codon frequencies were normalized to probabilities. Processed raw RNA sequencing Star counts were obtained from the Digital Expression Explorer V2 database (http://dee2.io/index.html)[23] and filtered for experiments that passed quality control. Raw mRNA data were transformed to transcripts per million (TPM) counts[24] and genes with zero mRNA output (TPM < 5) were removed. Prior to modeling, the mRNA counts were Box-Cox transformed.[25]

Yeast pangenome and quantitative traits dataset

The gene presence/absence (P/A) encoding of *S. cerevisiae* pan genome were obtained from Li et al.[26] The 35 quantitative traits were obtained from Peter et al.[27]

Monte Carlo simulations on expected R^2 score

Given the true functions between features and labels $f(x)$:

1. Randomly generate 1000 samples from $N(0,1)$ as x. Then true values are $y = f(x)$;
2. Randomly generate a noise vector ε_y. Each $\varepsilon_{y,i}$ is randomly sampled from $N(0,\sigma_{y,i}^2)$, where $\sigma_{y,i}^2$ is randomly sampled from $\chi^2(1)$;
3. $y_{obs} = y + \varepsilon_y$;
4. Add noise to $x_{obs} = x + \varepsilon_x$, in which ε_x is sampled from a normal distribution with zero-mean and variance of σ_x^2 (varying from 0 to 1);
5. Calculate R_{ML}^2 by performing a 2-fold cross-validation on dataset $\{x_{obs,i}, y_{obs,i}\}$ with support vector machine regression model (another inner 2-fold cross-validation for hyper-parameter optimization);
6. Calculate upper bound for R^2 with $\left\langle R^2 \right\rangle_{LG} = \frac{\sigma_{obs}^2 - \overline{\sigma_y^2}}{\sigma_{obs}^2}$ and $\left\langle R^2 \right\rangle_{FP} = \frac{\sigma_{obs}^2 - 2\overline{\sigma_y^2}}{\sigma_{obs}^2}$, where σ_{obs}^2 is the variance of y_{obs} and $\overline{\sigma_y^2}$ is the average value of randomly generated $\sigma_{y,i}^2$.
7. Repeat steps 1 to 6 for 1000 times.

A linear function $f(x) = 2x + 1$ and a nonlinear function $f(x) = 2sin(x) + 1$ were tested, respectively.

Monte Carlo simulations on data cleaning

Define a linear function $f(x) = \sum_{i=1}^{10} x_i$ as the true function to map 10 features to a target y. Each feature follows a standard normal distribution.

1. Randomly generate feature of 1000 samples as X. Calculate real target values y;

2. Randomly generate a noise vector ε_y. Each $\varepsilon_{y,i}$ is randomly sampled from $N(0,\sigma_{y,i}^2)$, where $\sigma_{y,i}^2$ is randomly sampled from $\chi^2(5)$;
3. Calculate observed target values via $y_{obs} = y + \varepsilon_y$, and resulted a dataset $\{X, y_{obs}\}$;
4. Assume we only know the first n features $(n = 2,4,6,8,10)$, Sort all samples based on $\sigma_{y,i}^2$ values, gradually remove the samples with the highest $\sigma_{y,i}^2$ values, calculate R^2 score of a linear function via a 2-fold cross validation on such a dataset with only a subset of features.
5. Repeat steps 1 through 4 for a total of 100 times.

Feature extraction for enzymes in T_{opt} dataset

The 5494 features from iFeature were broken up into 20 subfeature sets according to their types, and their predictive power was evaluated using 5 different regression models (linear, elastic net, Bayesian ridge, decision tree and random forest regressors).

A total of 20 different sets of protein features were extracted with iFeature[28] using default settings: amino acid composition (AAC, 20 features), dipeptide composition (DPC, 400), composition of *k*-spaced amino acid pairs (CKSAAP, 2400), dipeptide deviation from expected mean (DDE, 400), grouped amino acid composition (GAAC, 5), composition of *k*-spaced amino acid group pairs (CKSAAGP, 150), grouped dipeptide composition (GDPC, 25), grouped tripeptide composition (GTPC, 125), Moran autocorrelation (Moran, 240), Geary autocorrelation (Geary, 240), normalized Moreau-Broto (NMBroto, 240), composition-transition-distribution (CTDC, 39; CTDT, 39; CTDD, 195), conjoint triad (CTriad, 343), conjoint *k*-spaced triad (KSCTriad, 343), pseudo-amino acid composition (PAAC, 50), amphiphilic PAAC (APAAC, 80), sequence-order-coupling number (SOCNumber, 60) and quasi-sequence-order descriptors (QSOrder, 100). In total, we obtained 5494 features from iFeature. Furthermore, we additionally obtained features in the form of sequence embedding representations encoded by a deep learning model UniRep,[29] which is a Multiplicative Long-Short-Term-Memory (mLSTM) Recurrent Neural Networks (RNNs) that was trained on the UniRef50 dataset.[30] A total of 1900 × 3 features were extracted for each protein sequence using UniRep.

Supervised classical ML methods

Input features were first scaled to a standard normal distribution by $x_{N,i} = \frac{x_i - u_i}{\sigma_i}$, where x_i is the values of feature i of all samples, u_i and σ_i are the mean and standard deviation of x_i, respectively. Two linear regression algorithms BayesianRidge and Elastic Net as well as three non-nonlinear algorithms Decision Tree, Random Forest and Support Vector Machine[6] were evaluated on each feature set (iFeatures and UniRep). The evaluation was conducted via a nested cross-validation approach: an inner 3-fold cross validation was used for the hyper-parameter optimization via a grid-search strategy and an outer 5-fold cross-validation was used to estimate the model performance (see Table S2 for hyper-parameter values). With the transcriptomics data, linear regression was the only algorithm used, as it was previously found to outperform all other algorithms with a similar dataset.[5] For genomics datasets, only the random forest regression was tested. All ML analysis was performed with scikit-learn (v0.20.3)[31] using default settings and Python v3.6.7.

Supervised deep ML methods

To test the performance of a deep neural networks on the prediction of enzyme T_{opt}, architectures were tested that comprised up to 9 convolutional neural network (CNN) layers[32] followed by 2 fully connected (FC) layers.[33] Batch normalization[34] and weight dropout[35] were applied after all layers and max-pooling[36] after CNN layers. The Adam optimizer[37] with MSE loss function and ReLU activation function[38] with uniform[7] weight initialization were used. In total, 26 hyperparameters were optimized over a predefined parameter space (Table S3) using a tree-structured Parzen estimators approach[39] at default settings for 1000 iterations.[40,41] The Keras v2.2 and Tensorflow v1.10 software packages were used.

Prediction of T_{opt} for enzymes from BRENDA and CAZy

Sequences and associated OGT values for the BRENDA database was obtained from Li et al.[42] For the CAZy database, enzyme information including protein name, EC number, Organism, GenBank id, Uniprot id, PDB id and CAZy family id were obtained from http://www.cazy.org/.[43] 1 346 471 proteins with unique GenBank identifiers were obtained. Protein sequences were first downloaded from NCBI ftp site: https://ftp.ncbi.nih.gov/ncbi-asn1/protein_fasta/. Then only those sequences that were present in the CAZy dataset were kept by matching GenBank identifier. 924 642 sequences could be mapped to an OGT value by cross-referencing the source organism name and an OGT dataset.[18] Only the species names were checked, ignoring strain designations, for instance *S. cerevisiae* S288 C was considered as *S. cerevisiae*. For T_{opt} prediction on the BRENDA and CAZy data, the model with the best performance was selected, which in this case was the random forest model trained only on amino acid frequencies and OGT. The model was then trained on all samples in the training dataset. For the prediction, (1) the 20 amino acid frequencies were extracted with iFeature[28] and OGT values of their source organisms were mapped; (2) all these 21 features were normalized by subtracting the mean and then divided by the standard deviation obtained from the training dataset; and (3) these data were used as input of the model for the prediction of the T_{opt} values.

Theoretical Analysis

Starting from first principles, we mathematically derived a method to estimate upper bounds of model performance on holdout data in terms of R^2 and benchmarked the upper bound estimates against existing methods.

Estimating the theoretical upper bound of regression model performance

Given a set of samples with experimentally determined labels $\{y_{obs,i}\}$ and corresponding unknown real labels $\{y_i\}$, we assume a normally distributed experimental noise $\varepsilon_{y,i} \sim N(0,\sigma_{y,i})$: $y_{obs,i} = y_i + \varepsilon_{y,i}$ ($y_i \in R$), and that a complete set of features is known as $x_i \in R^k$ for each sample. This complete set of features can be used to 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) is a common metric to assess model performance and thus the R^2 of the model in the above argument is given by

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

where m is the number of samples. Although it is not possible to obtain an exact value from the above equation, since the real values $f(x_i)$ are unknown, we can instead obtain the expectation of R^2 (Note S1), which is given by

$$\left\langle R^2 \right\rangle = 1 - \left\langle \frac{\sum_{i=1}^{m}\left(y_{obs,i} - f\left(x_i\right)\right)^2}{\sum_{i=1}^{m}\left(y_{obs,i} - \overline{y}_{obs}\right)^2} \right\rangle = 1 - \frac{m}{m-3}\frac{\overline{\sigma_y^2}}{\sigma_{obs}^2} \qquad (2)$$

and in which $\overline{\sigma_y^2} = \frac{1}{m}\sum_{i=1}^{m}\sigma_{y,i}^2$. As the number of examples in ML is usually very large ($m >> 1$), we can approximate the final equation for upper bound estimation as

$$\left\langle R^2 \right\rangle \approx 1 - \frac{\overline{\sigma_y^2}}{\sigma_{obs}^2} = \frac{\sigma_{obs}^2 - \overline{\sigma_y^2}}{\sigma_{obs}^2} \qquad (3)$$

We refer to this upper bound estimate as $\left\langle R^2 \right\rangle_{LG}$ hereafter. It has a variance of $\frac{2m\left(m-2\right)}{\left(m-3\right)^2\left(m-5\right)}\frac{\overline{\sigma_y^4}}{\sigma_{obs}^4}$ (Note S1). Similarly, the best MSE that one can expect is given as $\left\langle MSE \right\rangle = \overline{\sigma_y^2}$ with a variance of $\frac{2\overline{\sigma_y^4}}{m}$ (Note S2). $\left\langle R^2 \right\rangle_{LG}$ gives a measure of the best R^2 one can expect when testing an ML model on a held out dataset.

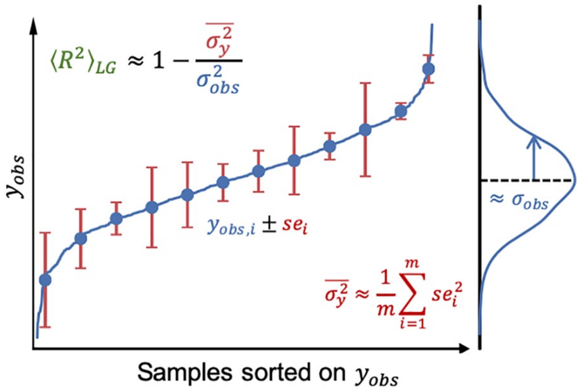

$\langle R^2 \rangle_{LG} \approx 1 - \frac{\overline{\sigma_y^2}}{\sigma_{obs}^2}$
y_{obs}
$y_{obs,i} \pm se_i$
$\approx \sigma_{obs}$
$\overline{\sigma_y^2} \approx \frac{1}{m}\sum_{i=1}^{m} se_i^2$
Samples sorted on y_{obs}

Figure 1. Schematic diagram depicting the estimation of the upper bound of model performance $\left\langle \boldsymbol{R}^2 \right\rangle_{LG}$ based on experimental label noise. $\underline{\sigma_y^2}$ can be approximated from the standard errors (*se*) of samples in the dataset, and σ_{obs}^2 can be approximated as the variance of the target values. Data shown were randomly generated, se_i denotes standard error of sample *i*. $\left\langle R^2 \right\rangle_{LG}$ is the expected upper bound for the coefficient of determination R^2 derived in this study (see section "Estimating the theoretical upper bound of regression model performance").

The new upper bound estimate $\left\langle R^2 \right\rangle_{LG}$, which is the R^2 expectation of the best possible ML model on holdout data, solely depends on 2 properties of the dataset: (1) the true variance of the observed response values (σ_{obs}^2) and (2) the average variance of experimental noise of all samples ($\overline{\sigma_y^2}$). In practice, σ_{obs}^2 and $\overline{\sigma_y^2}$ are unknown and have to be approximated from the dataset. $\sigma_{y,i}$ can be approximated with the standard error (SE) of n replicates, which represent the standard error of the mean, and σ_{obs}^2 can be approximated as the variance of the target values (Figure 1). Since the resulting $\left\langle R^2 \right\rangle_{LG}$ is an expectation and relies on approximated values, it does not strictly represent an upper bound for the R^2 of regression models and the real value may be slightly higher or lower. In this way the $\left\langle R^2 \right\rangle_{LG}$ estimate is analogous to using human-level performance to approximate upper bounds in image classification applications.[44-47]

$\left\langle R^2 \right\rangle_{LG}$ *upper bound estimates outperform existing methods*

In recent publications it has been proposed that, given a set of experimentally measured values $y_{obs,i}$, the best possible model is $y = x$ in which x are the values collected from another set of experiments conducted at identical conditions.[15,16] Under

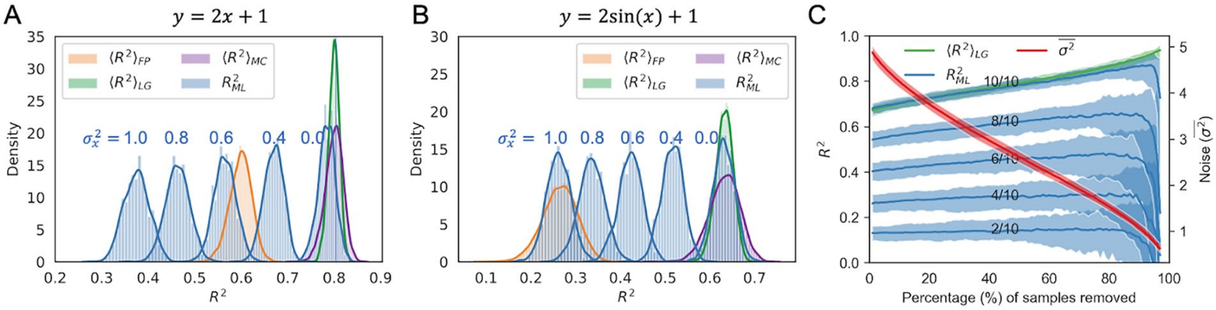


Figure 2. Monte Carlo simulations of the upper bound of R^2 assuming different levels of feature noise. $\left\langle R^2\right\rangle_{FP}$ and $\left\langle R^2\right\rangle_{LG}$ are expected upper bounds for the coefficient of determination R^2 with equations derived by Benevenuta and Fariselli et al[15] and this study, respectively. $\left\langle R^2\right\rangle_{MC}$ is the expected upper bound for R^2 obtained by Monte Carlo simulation as described in the "Results" section. R^2_{ML} is the R^2 obtained via a 2-fold cross-validation with a support vector machine. Two real functions were tested; (A) linear and (B) nonlinear. σ_x^2 is the variance of noise added to feature vector x, with examples of the observed data distributions depicted in Figure S1. (C) Monte Carlo simulation on data cleaning via gradually removing the samples with the largest $\sigma_{y,i}$. n/10 indicates that n features out of a complete set of 10 features were used to train and validate the model. Noise values are given as the average variance of all samples $(\overline{\sigma^2})$.

this assumption, the expectation of the upper bound for *MSE* is $2\overline{\sigma_y^2}$ and R^2 is $\frac{\sigma_{DB}^2 - \overline{\sigma_y^2}}{\sigma_{DB}^2 + \overline{\sigma_y^2}}$, where $\overline{\sigma_y^2}$ is the average variance of all sample noise and σ_{DB}^2 is the variance of the real values (without noise). Since $\sigma_{DB}^2 + \overline{\sigma_y^2} \approx \sigma_{obs}^2$, the upper bound for $\left\langle R^2\right\rangle$ becomes $\frac{\sigma_{obs}^2 - 2\overline{\sigma_y^2}}{\sigma_{obs}^2}$, and we refer to this upper bound as $\left\langle R^2\right\rangle_{FP}$ hereafter. Despite claims that no ML model could perform better than this upper bound,[15,16] by comparing the equations for $\left\langle R^2\right\rangle_{LG}$ and $\left\langle R^2\right\rangle_{FP}$, it is clear that $\left\langle R^2\right\rangle_{FP}$ estimates are lower than both $\left\langle R^2\right\rangle_{LG}$ estimates as well as achieved ML model performance.

To directly compare $\left\langle R^2\right\rangle_{FP}$ and $\left\langle R^2\right\rangle_{LG}$, we performed Monte Carlo simulations. Briefly, a random dataset $\{x_i, y_{\text{obs,i}}\}$ was generated from a known real function $f(x)$ with added experimental noise $\sigma_{y,i}$ (see section "Monte Carlo simulations on expected R^2 score"). For this dataset, $\left\langle R^2\right\rangle_{FP}$ and $\left\langle R^2\right\rangle_{LG}$ were calculated, and then the R^2 of a support vector machine regression model[48] trained on the data was calculated via a 2-fold cross validation approach (R^2_{ML}). This process was repeated for 1000 iterations. A linear (Figure 2A) and nonlinear real function (Figure 2B) were used in two separate simulations (see observed data distributions in Figure S1). Furthermore, to evaluate the effects of feature noise on regression model performance, we generated noise-free features ($\sigma_x^2 = 0.0$), as well as features with different levels of noise associated with them ($\sigma_x^2 = 0.2$, $\sigma_x^2 = 0.4$, $\sigma_x^2 = 0.6$, $\sigma_x^2 = 0.8$, $\sigma_x^2 = 1.0$). The simulations illustrated three key points. First, in both linear and nonlinear cases, R^2_{ML} is always smaller than or close to $\left\langle R^2\right\rangle_{LG}$, which confirms that $\left\langle R^2\right\rangle_{LG}$ gives a good estimation of the model performance upper bound. Second, the simulations show that there are ML models with R^2_{ML} higher than $\left\langle R^2\right\rangle_{FP}$, which is contrary to the expectation if $\left\langle R^2\right\rangle_{FP}$ is a true upper bound.[15,16] Third, as σ_x^2 increases, the ML model performance falls short of the R^2 upper bound, eventually falling below $\left\langle R^2\right\rangle_{FP}$. This shows that $\left\langle R^2\right\rangle_{LG}$ gives a more accurate estimation of the upper bound for the performance of ML models at any condition, including cases with or without noisy features. $\left\langle R^2\right\rangle_{FP}$ is however useful as an estimate of the reproducibility of experiments, in accordance with the assumptions in the original papers.[15,16]

Since x is normally distributed in the simulations (Figure 2B), $2sin(x)+1+\epsilon$ gives a non-normal distribution for y_{obs} values. While $\left\langle R^2\right\rangle_{LG}$ was derived under the assumption that y_{obs} is normally distributed, the Monte Carlo simulations indicate that $\left\langle R^2\right\rangle_{LG}$ is accurate also when applied to non-normal distributions (Figure 2B). Since it is challenging to prove this mathematically, we devised an additional simulation strategy to further test whether $\left\langle R^2\right\rangle_{LG}$ can be applied to a given dataset with non-normal distributed y_{obs} values given the experimental noise. Following the same notations as "Estimating the theoretical upper bound of regression model performance," we can get $y_i = y_{obs,i} - \varepsilon_{y,i}$, in which $\varepsilon_{y,i} \sim N(0, \sigma_{y,i})$. We can randomly draw a $\hat{y}_i$ as an estimation of y_i in this equation. The R^2 resulting from using $\{\hat{y}_i\}$ as the prediction of $\{y_{obs,i}\}$ is an independent estimation of the best achievable R^2. By repeating this step, a list of R^2 scores can be calculated and the average R^2 gives an estimation of the expectation of the best R^2 we can get (referred to as $\left\langle R^2\right\rangle_{MC}$). Since $\left\langle R^2\right\rangle_{MC}$ does not rely on the normality assumption of y_{obs} values, it can be applied to any distribution. As shown in Figure 2A and B, $\left\langle R^2\right\rangle_{LG}$ gives a consistent estimation of $\left\langle R^2\right\rangle_{MC}$, at least in the 2 tested cases (Figure 2A and B). With a dataset with non-normal distributed y_{obs} values, one can use this simulation strategy to obtain a $\left\langle R^2\right\rangle_{MC}$ to check if $\left\langle R^2\right\rangle_{LG}$ gives a correct estimation.

In the above analysis, idealized conditions were used in that all features were known. Conversely, in real-world ML

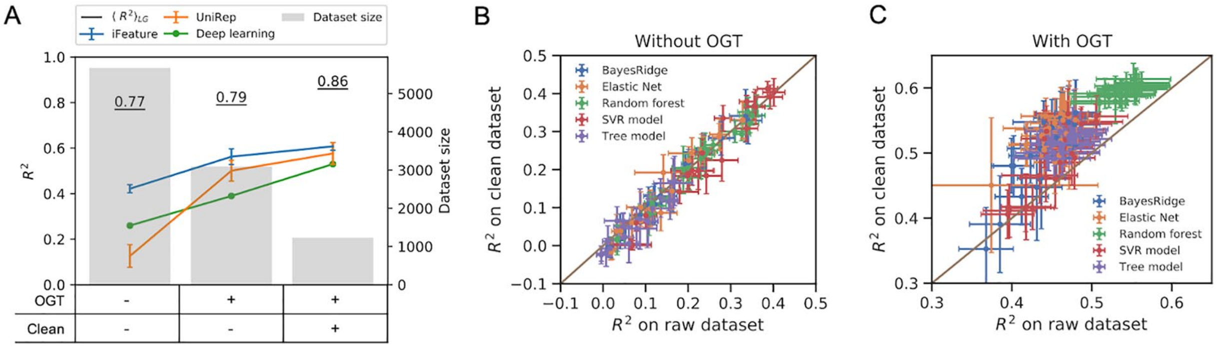


Figure 3. Development of machine learning models for the prediction of enzyme optimal temperature (T_{opt}). (A) Performance of classical models using iFeatures[28] and UniRep encoding[29] feature sets as well as a deep neural networks with one-hot encoded protein sequence as input. (B, C) Comparison of model performance on raw and clean dataset (B) with; and (C) without optimal growth temperature (OGT) as one of the features. The features calculated by iFeature were grouped into 20 sub-feature sets as described in the "Methods" section. Error bars show the standard deviation of R^2 scores obtained in 5-fold cross validation. $\langle R^2\rangle_{LG}$ is the expected upper bound for the coefficient of determination R^2 derived in this study (see section "Estimating the theoretical upper bound of regression model performance").

applications, typically only an incomplete set of features is known. To more accurately simulate this real-world situation, we performed Monte Carlo simulations using incomplete feature sets (see section "Monte Carlo simulations on data cleaning"). We also evaluated how ML model performance is affected by removal of the most noisy sample labels (Figure 2C). As anticipated, models trained with the full feature set (10/10) outperformed those trained with a subset of features, with the model containing all 10 features reaching the $\langle R^2\rangle_{LG}$ (Figure 2C). Furthermore, model performance generally improved as noisy samples were removed. However, an interesting observation is that the degree to which the models improve upon removal of noisy samples depends on how many features were used to train them. For instance, if only a small fraction of relevant features were used (2/10 in Figure 2C), the removal of the most noisy samples did not improve model performance. In contrast, when a majority of the relevant features were known (8/10 and 10/10 in Figure 2C), the removal of noisy samples significantly improved the model performance in terms of R^2. These results indicate that when R^2_{ML} is very far from the $\langle R^2\rangle_{LG}$ upper bound, model performance can be most readily improved by obtaining additional or more relevant features, as opposed to performing data cleaning to reduce sample noise.

Experimental Case Studies

We next explored the applicability of $\langle R^2\rangle_{LG}$ to inform ML model development on real-world data that included enzyme optimal temperatures, transcriptomic and genomic datasets.

Using the theoretical upper bound to inform modeling

We first tackled the problem of obtaining models to accurately predict enzyme optimal catalytic temperatures (T_{opt}) using features extracted from their protein primary structures. A dataset comprising the T_{opt} of 5343 individual enzymes was collected from the BRENDA[17] database. Here, using enzymes for which T_{opt} values had been measured in multiple experiments, the experimental noise $\overline{\sigma_y^2}$ was estimated as $(7.84\,°C)^2$ and σ_{obs}^2 was $(16.32\,°C)^2$, giving an $\langle R^2\rangle_{LG}$ upper bound of 0.77. Moreover, an estimated $\langle R^2\rangle_{MC}$ of 0.77 confirmed that $\langle R^2\rangle_{LG}$ could be applied on this dataset with non-normally distributed T_{opt} values (Figure S2). To provide features for ML model training (see section "Supervised classical ML methods"), two established feature extraction methods were applied to the protein primary structures, one based on domain knowledge (iFeature,[28] 5494 features) and the other based on embeddings obtained from unsupervised deep learning (UniRep,[29] 5700 features). Despite testing six different types of regression algorithms to predict enzyme T_{opt} using the two feature sets, even the best achieved R^2 of 0.42 (average over 5-fold cross-validations, Figure 3A) was only 55% of the estimated $\langle R^2\rangle_{LG}$ upper bound, indicating that the model could be further improved. Such improvement might be achieved through either feature engineering or noise reduction, as seen in the Monte Carlo simulations (Figure 2C).

First, we performed feature engineering by including the OGT of the organism, from which the enzyme was derived, as an additional feature into the iFeature and UniRep feature sets. Consequently, the dataset size decreased to 2917 enzymes, as 55% of the samples were omitted due to unknown OGT values of their source organisms. This led to models improved by 33% and 384%, respectively (Figure 3A), and the best resulting R^2 (0.56) achieved 71% of the estimated $\langle R^2\rangle_{LG}$ (0.79). These results are consistent with our previous work,[42] where it was shown that prediction of enzyme T_{opt} was significantly improved when including OGT as a feature. We then tested whether a deep convolutional neural network (Figure S3, see section "Supervised deep ML methods") could discover additional features in the enzyme primary structures[5,40] and provide better predictive models (with/without OGT as an additional feature). This did not, however, lead to models that outperformed the classical ones trained on iFeatures (Figure 3A). Next, we considered reducing the noise $(\overline{\sigma_y^2})$ in $\mathrm{T_{opt}}$ values as a means to further improve model performance and used the information from the "comment" field associated with each enzyme in the BRENDA database to remove values that were deemed less likely to represent true catalytic optima (see section "Enzyme catalytic temperature optima dataset"). Despite dramatically reducing the number of samples in the dataset (1902 enzymes of which 1232 with known OGT), the experimental noise $\overline{\sigma_y^2}$ was reduced from $(7.84\,°C)^2$ to $(7.22\,°C)^2$ and the calculated $\langle R^2 \rangle_{LG}$ increased from 0.79 to 0.85 ($\langle R^2 \rangle_{MC}$ was 0.85). Accordingly, the best model obtained with this dataset achieved an improved R^2 of 0.61, which again was around 71% of $\langle R^2 \rangle_{LG}$. Here, in accordance with the expectation that large training datasets are required for optimal deep learning performance,[40] the convolutional network displayed a reduced R^2 score (Figure 3A). Finally, further in-depth analysis to explore how different sub-features of the iFeature set contributed to predictive accuracy (20 sub-feature sets used, see section "Feature extraction for enzymes in $\mathrm{T_{opt}}$ dataset") showed that each sub-feature set only improved model performance when OGT was included as an additional feature (Figures 3B, C and S4, 5). This is consistent with Monte Carlo simulation results showing that noise reduction is mainly beneficial with more complete feature sets (Figure 2C).

As a side note, we used the improved model (a random forest trained on amino acid composition and OGT) to update $\mathrm{T_{opt}}$ annotation of BRENDA enzymes in the Tome package[42] and also extended it to carbohydrate-active (CAZy) enzymes[43] (Figure S6) (https://github.com/EngqvistLab/Tome).

Further analyses of $\langle R^2 \rangle_{LG}$ in relation to experimental noise

We next explored how the amount of experimental noise in the response variables can affect the $\langle R^2 \rangle_{LG}$ and model performance. A suitable problem for this was the prediction of intrinsic gene expression levels in *S. cerevisiae*, since thousands of RNAseq experiments across multiple conditions are available for this species.[23] For a given gene, the intrinsic expression level was defined as the average mRNA level across the different experiments and conditions.[5] The noise level could then be adjusted by increasing or decreasing the number of sampled data points (i.e. replicates) (Figure 4 inset), where the corresponding standard deviation was used to quantify the amount of noise present within the intrinsic expression levels (see section "Protein transcription level dataset"). Apart from estimating the $\langle R^2 \rangle_{LG}$ upper bounds, the achievable predictive performance was explored by building linear regression models

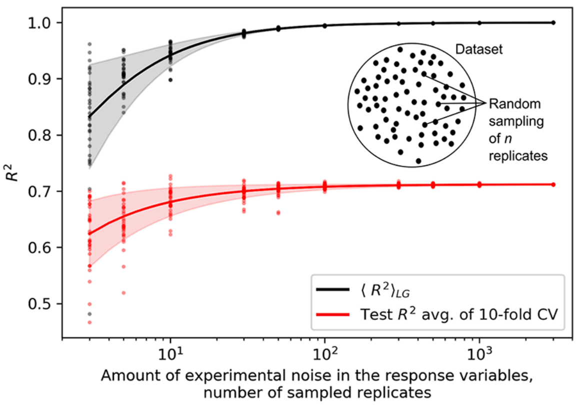
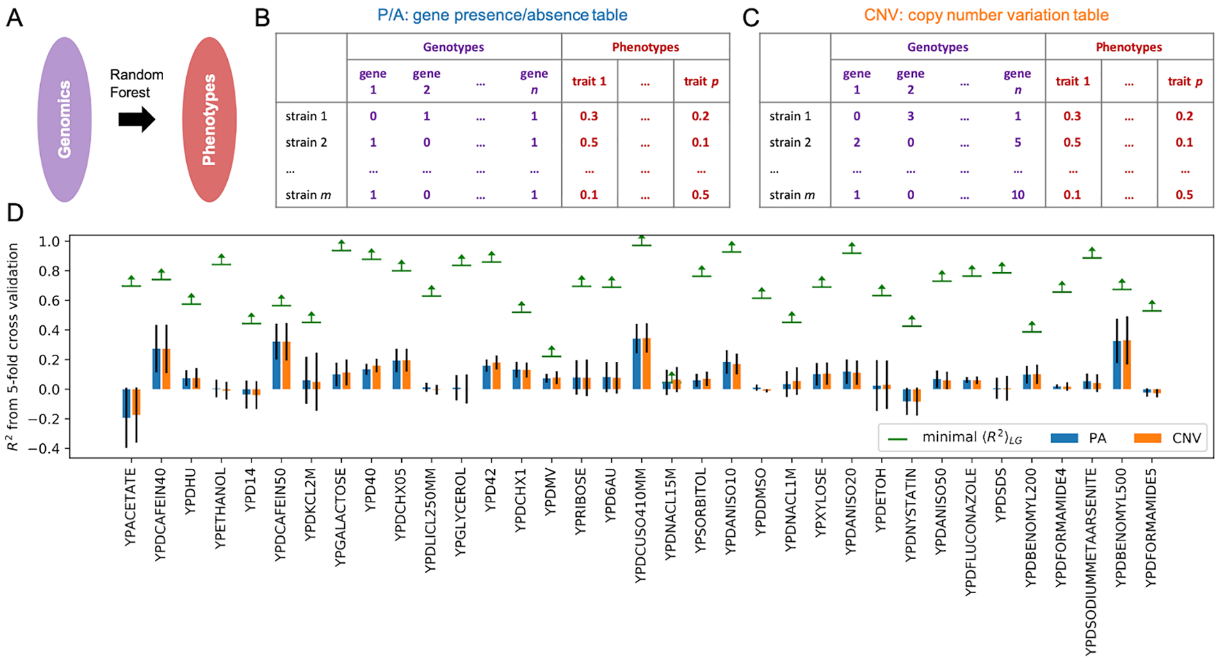


Figure 4. Amount of experimental noise affects estimates of $\langle R^2 \rangle_{LG}$ and model performance. Analysis of the effect of experimental noise in the response variables on the $\langle R^2 \rangle_{LG}$ upper bound estimates (black) and predictive performance of ML models (red) with the case of a large yeast multi-experiment transcriptomics dataset.[23] The noise level was varied by adjusting the number of data replicates with random sampling (inset figure). Lines and shaded areas depict means and standard deviations of the 30 measurements per each *n* replicates, depicted as points. $\langle R^2 \rangle_{LG}$ is the expected upper bound for the coefficient of determination R^2 derived in this study (see section "Estimating the theoretical upper bound of regression model performance"), CV denotes cross validations.

using DNA sequence features (codon usage)[5,49] as input. We observed a strong effect of the amount of experimental noise on the theoretical upper bound, especially with a smaller number of data replicates (Figures 4 and S7). Similarly, the variability of the $\langle R^2 \rangle_{LG}$ upper bound also markedly decreased with an increasing number of replicates. Therefore, with an insufficient amount of replicates, apart from a lower confidence in the estimated upper bound, the variability of the data was found to also drastically affect the predictive performance and accuracy of the models. This suggests that for data that are inherently noisy, such as those obtained from transcriptomics, the $\langle R^2 \rangle_{LG}$ upper bound, as well as the overall ML performance, can both be improved by increasing the number of experimental replicates generated for downstream computational analysis.[50] For accurate condition-specific or cross-condition modeling, the number of replicates of at least 100, with most reliable results above 1000, should be used (Figures 4 and S6). Such dataset sizes are nowadays highly feasible, especially in the case of genomics, transcriptomics and proteomics data, as resources that comprise thousands of experiments are readily available for each model organism.[23,26,27]

Finally, for some datasets it may not be feasible to estimate the experimental noise, for instance, if the values for replicates in an experiment are not available. We thus analyzed how $\langle R^2 \rangle_{LG}$ can be used to define the predictive potential of biological regression analysis even in the absence of direct experimental noise estimates. Since $\langle R^2 \rangle_{LG}$ is an upper bound estimate, $R_{ML}^2 \le \langle R^2 \rangle_{LG}$ holds true, from which we obtain that $\overline{\sigma_y^2} \le (1 - R_{ML}^2) \times \sigma_{obs}^2$. If multiple datasets with the same



	Genotypes				Phenotypes		
	gene 1	gene 2	...	gene *n*	trait 1	...	trait *p*
strain 1	0	1	...	1	0.3	...	0.2
strain 2	1	0	...	1	0.5	...	0.1
...	...	...	...	...	...	...	...
strain *m*	1	0	...	1	0.1	...	0.5

	Genotypes				Phenotypes		
	gene 1	gene 2	...	gene *n*	trait 1	...	trait *p*
strain 1	0	3	...	1	0.3	...	0.2
strain 2	2	0	...	5	0.5	...	0.1
...	...	...	...	...	...	...	...
strain *m*	1	0	...	10	0.1	...	0.5

Figure 5. $\langle R^2\rangle_{LG}$ is applicable even in case the experimental noise is unknown. Analysis of 34 quantitative traits of *S. cerevisiae* from its pan-genome composition. (A) A random forest model applied to predict yeast phenotypes from genomics features. Genomes are represented as (B) gene presence/absence table and (C) copy number variance table in the pangenome.[26] (D) Obtained R^2 score for 35 different phenotypes. Experimental trait values were taken from Peter et al.[27] Detailed label description can be found in Table S2 of Peter et al.[27] Error bars show the standard deviation of R^2 scores obtained in 5-fold cross validation. $\langle R^2\rangle_{LG}$ is the expected upper bound for the coefficient of determination R^2 derived in this study (see section "Estimating the theoretical upper bound of regression model performance").

level of (unknown) experimental noise are available, the inequality holds for all datasets, meaning that $\overline{\sigma_y^2} \le min(\{(1-R^2_{ML,i})\times\sigma^2_{obs,i} \mid i=1,..,s\})$, in which s is the number of the datasets. In this way it is possible to estimate the maximal level of experimental noise based on the ML results, and then further use it to obtain the minimal value of $\langle R^2\rangle_{LG}$, where $\langle R^2\rangle_{LG}$ could be any value between $\langle R^2\rangle_{LG,min}$ and 1.0. This idea was applied on a problem to predict yeast phenotypes directly from genomes[51] (Figure 5A) using a dataset of growth profiles of 971 sequenced *S. cerevisiae* isolates under 35 stress conditions,[27] where experimental noise was not reported nor replicate data provided (see section "Yeast pangenome and quantitative traits dataset"). To analyze these data we made use of a published *S. cerevisiae* pan-genome[26] that included all protein-coding genes across 971 isolates with measured phenotypes. Here, the pan-genome was represented as either a gene presence/absence table (P/A, Figure 5B), or copy number variation table (CNV) which contains additional information to P/A (Figure 5C). Using these features we could estimate $\langle R^2\rangle_{LG,min}$ for each condition by training a random forest regressor on the 35 different quantitative traits. P/A and CNV showed a similar predictive power and could explain at most 30% of the variance (Figure 5D: R^2 was ~0.3) for some traits like the growth profile under the YPD medium enriched with 40 mM of caffeine (YPDCAFEIN40). Furthermore, with R^2_{ML} s for these 35 datasets, the maximal experimental noise were estimated as $\overline{\sigma_y^2} \le 0.076^2$, based on which we could finally estimate the $\langle R^2\rangle_{LG,min} \approx 1-\frac{0.076^2}{\sigma^2_{obs,i}}$ for each condition (Figure 5C). Since most of the traits did not follow a normal distribution (Figure S8), $\langle R^2\rangle_{MC}$ was obtained by Monte Carlo simulation with $\overline{\sigma_y^2}$ of 0.076^2 for each dataset and was used to cross-check the $\langle R^2\rangle_{LG}$ values, indicating that breaking the normality assumption did not adversely affect the R^2 estimate (Figure S9 $:\langle R^2\rangle_{MC}$ was consistent with $\langle R^2\rangle_{LG}$). Despite that for a small number of traits (e.g. YPDNACL15M) the low $\langle R^2\rangle_{LG,min}$ was too low to be useful (Figure 5D), in most cases it was higher than the current predictive performance of our models (e.g. > 0.97 with YPDCUSO410MM). Thus, for most of the conditions, the estimated upper bounds showed great potential for further improvement of model performance (Figure 5D), demonstrating the applicability and usefulness of $\langle R^2\rangle_{LG}$ even in case the experimental noise is unknown.

Discussion

In the present study, we establish a method to estimate an upper bound for expected ML regression model performance on holdout data. This addresses an important need in the ML field as human performance on multi-dimensional data is poor and cannot be used to bootstrap regression model performance,[9,10] an approach that is common when developing ML systems for image analysis.[7,8] The coefficient of determination upper bound (model performance) for regression analysis is: $\left\langle R^2 \right\rangle_{LG} \approx 1-\frac{\overline{\sigma_y^2}}{\sigma_{obs}^2}$, and depends only on the experimental noise $\overline{\sigma_y^2}$ and the variance of observed labels σ_{obs}^2 (Figure 1), under the assumptions that observed label values $\{y_{obs,i}\}$ are normally distributed and that each $y_{obs,i}$ has a normally distributed experimental noise with 0 mean. With non-normally distributed $\left\{y_{obs,i}\right\}$, we provide a Monte Carlo based approach ($\left\langle R^2 \right\rangle_{MC}$) to estimate $\left\langle R^2 \right\rangle_{LG}$. In all tested cases, $\left\langle R^2 \right\rangle_{LG}$ gives results consistent with $\left\langle R^2 \right\rangle_{MC}$, indicating that even though it was derived under the normality assumption of $\left\{y_{obs,i}\right\}$, it can also be applied to data with other distributions (Figures 2A and B and S8). $\left\langle R^2 \right\rangle_{LG}$ is thus confirmed using Monte-Carlo simulations and also shown to outperform existing measures[15] (Figure 2A and B).

Our case studies demonstrate how calculating the $\left\langle R^2 \right\rangle_{LG}$ upper bound estimate for experimental data yields a more realistic modeling objective than naively assuming that an R^2 of 1.0 is possible. For instance, in the prediction of enzyme optimal temperature, $\left\langle R^2 \right\rangle_{LG}$ was estimated at 0.86 for a specific dataset (Figure 3A), and one should not expect to obtain ML models with R^2 scores on holdout data above this value. Moreover, achieving upper bound performance is only possible when a complete set of noise-free features relevant for the predicted labels are used for model training and prediction (Figure 2A to C). When noisy features are used, the performance attainable by ML algorithms will be lower than $\left\langle R^2 \right\rangle_{LG}$, and thus for classical ML models relying on engineered features, models with holdout data R^2 close to their upper bound are not easily achieved in practice (Figures 3A, 4 and 5D). On the other hand, if the estimated $\left\langle R^2 \right\rangle_{LG}$ upper bound for a specific problem and dataset is low (label values are noisy compared to the label variance, $\overline{\sigma_y^2}$ is close to σ_{obs}^2), it may not be worthwhile to attempt modeling at all, at least not before collecting additional data (more replicates). An example of this is the prediction of melting temperatures of human proteins (Table S1) using the dataset from Leuenberger et al.[52] The sample labels for human proteins in this dataset have a large level of noise ($\overline{\sigma_y^2}$, 5.49)^2 compared to the label variance (σ_{obs}^2, 6.57)^2 and the calculated $\left\langle R^2 \right\rangle_{LG}$ was therefore correspondingly low at approximately 0.30 (using equation (3), see section "Estimating the theoretical upper bound of regression model performance"). Even if a ML model with upper bound performance could be developed for these data, it would have little predictive value. In contrast, for three other, non-human, organisms in the Leuenberger dataset the calculated $\left\langle R^2 \right\rangle_{LG}$ was above 0.90, indicating that the development of predictive ML models may be worthwhile (Table S1).

To conclude, our method for estimating upper bounds for model performance on holdout data should aid researchers from diverse fields in developing ML regression models that reach their maximum potential.

Author Contributions

GL and MKME conceptualized the research. GL and JG mathematically derived the $\left\langle R^2 \right\rangle_{LG}$. GL performed Monte Carlo simulations. GL, JZ, JL and MKME analyzed and interpreted the results of predicting enzyme T_{opt}. JZ and AZ analyzed and interpreted the results of transcriptomics data. GL, BJ and JN analyzed or interpreted the genomics results. GL, JZ and MKME wrote the initial draft of the paper. GL, JZ, AZ, JN and MKME carried out revisions on the initial draft and wrote the final version.

Availability of Data and Materials

The tome package is available on GitHub (https://github.com/EngqvistLab/Tome/). The annotated T_{opt} values and sourceorganism OGTs for enzymes in the BRENDA and CAZy databases are available as flatfiles on Zenodo (https://zenodo.org/record/3578468#.XffgbpP0nOQ, DOI: 10.5281/zenodo.3578467). Other scripts and datasets are available on GitHub (https://github.com/EngqvistLab/Supplemetenary_scripts_datasets_R2LG).

ORCID iDs

Boyang Ji https://orcid.org/0000-0002-7269-4342
Martin KM Engqvist https://orcid.org/0000-0003-2174-2225

Supplemental Material

Supplemental material for this article is available online.

References

1. Zhou Y, Li G, Dong J, Xing XH, Dai J, Zhang C. MiYA, an efficient machine-learning workflow in conjunction with the YeastFab assembly strategy for combinatorial optimization of heterologous metabolic pathways in Saccharomyces cerevisiae. *Metab Eng*. 2018;47:294-302.
2. Romero PA, Krause A, Arnold FH. Navigating the protein fitness landscape with Gaussian processes. *Proc Natl Acad Sci USA*. 2013;110:E193-E201.
3. Zelezniak A, Vowinckel J, Capuano F, et al. Machine learning predicts the yeast metabolome from the quantitative proteome of kinase knockouts. *Cell Syst*. 2018;7:269-283.e6.
4. Zrimec J, Lapanje A. DNA structure at the plasmid origin-of-transfer indicates its potential transfer range. *Sci Rep*. 2018;8:1820.
5. Zrimec J, Börlin CS, Buric F, et al. Deep learning suggests that gene expression is encoded in all parts of a co-evolving interacting gene regulatory structure. *Nat Commun*. 2020;11:6141.
6. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer Science & Business Media; 2013.
7. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. Paper presented at: Proceedings of the IEEE International Conference on Computer Vision; December 7-13, 2015:1026-1034; Santiago, Chile. https://ieeexplore.ieee.org/document/7410480.

8. Liu X, Faes L, Kale AU, et al. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *Lancet Digit Health*. 2019;1:e271-e297.
9. Ripley BD. *Pattern Recognition and Neural Networks*. Cambridge, UK: Cambridge University Press; 2007.
10. Domingos P. A few useful things to know about machine learning. *Commun ACM*. 2012;55:78-87.
11. Botchkarev A. A new typology design of performance metrics to measure errors in machine learning regression algorithms. *IJIKM*. 2019;14:45-76.
12. Tsimring LS. Noise in biology. *Rep Prog Phys*. 2014;77:026601.
13. Harris EF, Smith RN. Accounting for measurement error: a critical but often overlooked process. *Arch Oral Biol*. 2009;54:S107-S117.
14. Bruggeman FJ, Teusink B. Living with noise: on the propagation of noise from molecules to phenotype and fitness. *Curr Opin Syst Biol*. 2018;8:144-150.
15. Benevenuta S, Fariselli P. On the upper bounds of the real-valued predictions. *Bioinform Biol Insights*. 2019;13:1177932219871263.
16. Montanucci L, Martelli PL, Ben-Tal N, Fariselli P. A natural upper bound to the accuracy of predicting protein stability changes upon mutations. *Bioinformatics*. 2019;35:1513-1517.
17. Jeske L, Placzek S, Schomburg I, Chang A, Schomburg D. BRENDA in 2019: a European ELIXIR core data resource. *Nucleic Acids Res*. 2019;47:D542-D549.
18. Engqvist MKM. Correlating enzyme annotations with a large set of microbial growth temperatures reveals metabolic adaptations to growth at diverse temperatures. *BMC Microbiol*. 2018;18:177.
19. Cherry JM, Hong EL, Amundsen C, et al. Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic Acids Res*. 2012;40:D700-D705.
20. Cherry JM, Adler C, Ball C, et al. SGD: Saccharomyces Genome Database. *Nucleic Acids Res*. 1998;26:73-79.
21. Xu Z, Wei W, Gagneur J, et al. Bidirectional promoters generate pervasive transcription in yeast. *Nature*. 2009;457:1033-1037.
22. Nagalakshmi U, Wang Z, Waern K, et al. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*. 2008;320:1344-1349.
23. Ziemann M, Kaspi A, El-Osta A. Digital expression explorer 2: a repository of uniformly processed RNA sequencing data. *GigaScience*. 2019;8:giz022.
24. Li B, Ruotti V, Stewart RM, Thomson JA, Dewey CN. RNA-seq gene expression estimation with read mapping uncertainty. *Bioinformatics*. 2010;26:493-500.
25. Box GEP, Cox DR. An analysis of transformations. *J R Stat Soc Series B Stat Methodol*. 1964;26:211-243.
26. Li G, Ji B, Nielsen J. The pan-genome of Saccharomyces cerevisiae. *FEMS Yeast Res*. 2019;19:foz064.
27. Peter J, De Chiara M, Friedrich A, et al. Genome evolution across 1,011 Saccharomyces cerevisiae isolates. *Nature*. 2018;556:339-344.
28. Chen Z, Zhao P, Li F, et al. iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*. 2018;34:2499-2502.
29. Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM. Unified rational protein engineering with sequence-based deep representation learning. *Nat Methods*. 2019;16:1315-1322. doi:10.1038/s41592-019-0598-1.
30. Suzek BE, Wang Y, Huang H, et al. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*. 2015;31:926-932.
31. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 2011;12:2825-2830.
32. LeCun Y, Haffner P, Bottou L, Bengio Y. Object recognition with gradient-based learning. In: Forsyth DA, Mundy JL, di Gesú V, Cipolla R, eds. *Shape, Contour and Grouping in Computer Vision*. Berlin, Germany: Springer; 1999:319-345.
33. Hou J, Adhikari B, Cheng J. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*. 2018;34:1295-1303.
34. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv*. 2015. https://arxiv.org/pdf/1502.03167.pdf.
35. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014;15:1929-1958.
36. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, eds. *Advances in Neural Information Processing Systems 25*. Red Hook, NY: Curran Associates, Inc.; 2012:1097-1105.
37. Kingma DP, Ba J. Adam: a method for stochastic optimization. *arXiv*. 2014. https://arxiv.org/pdf/1412.6980.pdf.
38. Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. Paper presented at: Proceedings of the 27th International Conference on Machine Learning (ICML-10); June 21-24, 2010:807-814; Haifa, Israel. https://www.cs.toronto.edu/~hinton/absps/reluICML.pdf.
39. Bergstra J, Komer B, Eliasmith C, Yamins D, Cox DD. Hyperopt: a Python library for model selection and hyperparameter optimization. *Comput Sci Discov*. 2015;8:014008.
40. Bengio Y. Practical recommendations for gradient-based training of deep architectures. In: Montavon G, Orr GB, Müller K-R, eds. *Neural Networks: Tricks of the Trade*. 2nd ed. Berlin, Germany: Springer; 2012:437-478.
41. Bergstra JS, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. In: Shawe-Taylor J, Zemel RS, Bartlett PL, Pereira F, Weinberger KQ, eds. *Advances in Neural Information Processing Systems 24*. Red Hook, NY: Curran Associates, Inc.; 2011:2546-2554.
42. Li G, Rabe KS, Nielsen J, Engqvist MKM. Machine learning applied to predicting microorganism growth temperatures and enzyme catalytic optima. *ACS Synth Biol*. 2019;8:1411-1420.
43. Lombard V, Golaconda Ramulu H, Drula E, Coutinho PM, Henrissat B. The carbohydrate-active enzymes database (CAZy) in 2013. *Nucleic Acids Res*. 2014;42:D490-D495.
44. Matek C, Schwarz S, Spiekermann K, Marr C. Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks. *Nat Mach Intell*. 2019;1:538-544.
45. Dodge S, Karam L. A study and comparison of human and deep learning recognition performance under visual distortions. Paper presented at: 2017 26th International Conference on Computer Communication and Networks (ICCCN); July 31-August 3, 2017; Vancouver, BC, Canada. doi:10.1109/icccn.2017.8038465.
46. Singh AV, Maharjan RS, Kanase A, et al. Machine-learning-based approach to decode the influence of nanomaterial properties on their interaction with cells. *ACS Appl Mater Interfaces*. 2021;13:1943-1955.
47. Singh AV, Rosenkranz D, Ansari MH, et al. Artificial intelligence and machine learning empower advanced biomedical material design to toxicity prediction. *Adv Intell Syst*. 2020;2:2000084.
48. Smola AJ, Schölkopf B. A tutorial on support vector regression. *Stat Comput*. 2004;14:199-222.
49. Cheng J, Maier KC, Avsec Rus ŽP, Gagneur J. Cis-regulatory elements explain most of the mRNA stability variation across genes in yeast. *RNA*. 2017;23:1648-1659.
50. Singh AV, Ansari MHD, Rosenkranz D, et al. Artificial intelligence and machine learning in computational nanotoxicology: unlocking and empowering nanomedicine. *Adv Healthc Mater*. 2020;9:e1901862.
51. Jelier R, Semple JI, Garcia-Verdugo R, Lehner B. Predicting phenotypic variation in yeast from individual genome sequences. *Nat Genet*. 2011;43:1270-1274.
52. Leuenberger P, Ganscha S, Kahraman A, et al. Cell-wide analysis of protein thermal unfolding reveals determinants of thermostability. *Science*. 2017;355:eaai7825.