Deviation Error: assessing predictions for replicate measurements in genomics and beyond

Husam Abdulnabi and J. Timothy Westwood

Department of Cell and Systems Biology, University of Toronto Mississauga, Canada

Correspondence: husam.abdulnabi@mail.utoronto.com

Abstract

A quantitative measurement can have variation, referred to here as measurement variation, which is a probability distribution. Machine Learning models typically produce a prediction corresponding to the mode of the measurement variation. The Deviation Error is a novel metric, described here, to assess predictions that accounts for measurement variation.

Measurement variations in genomics data were explored. Towards a general prescription for modelling genomics measurements to reduce the Deviation Error, different loss functions were used to fit models on synthetically generated data that mimics genomics measurements. Compared to using the Mean Squared Error as the loss function, none of the other loss functions examined yielded models that performed significantly better. However, using variants of the Mean Squared Log Error and the Negative Log Likelihood as loss functions yielded models that performed significantly worse.

1 Introduction

A measurement of a quantity can vary due to the variability of the measurement methods and the nature of the quantity itself (e.g. biological phenomena). The variation of a measurement, referred to here as measurement variation, is described by a probability distribution and is estimated by repeating the measurement, referred to here as repeat or replicate measurements.

The probability distribution for a discrete quantity, known as a probability mass function (PMF), describes the probability of observing a specific discrete quantity from real measurements. Conversely, the probability distribution for a continuous quantity, known as a probability density function (PDF), describes the relative likelihood of observing a specific continuous quantity from real measurements. A PDF can be used to determine the probability of observing a specific range of continuous values through integration. Notably, a probability distribution can be used to determine the probability of a value occurring relative to another (**Sup. Eq. 1**).

In the simplest case, Machine Learning (ML) models produce a prediction corresponding to the mode of the measurement variation, referred to here as the actual value, and assess the difference between the prediction and the actual value, known as the prediction error. Typically, the prediction error is the absolute difference of the prediction and the actual value, known as the absolute prediction error (**Eq. 1**) The absolute prediction error is used within larger functions such as the Mean Squared Error (**Eq. 2**) and the closely related Root Mean Squared Error (RMSE) (**Eq. 3**).

Eq. 1. Absolute Prediction Error

$$V(\hat{y},y) = |\hat{y} - \mathrm{y}|$$

where:

$y =$ actual value, $y \in Y$

$\hat{y} =$ predicted value, $\hat{y} \in \hat{Y}$

Eq. 2. Mean Squared Error

$$M(\hat{Y},Y) = \frac{1}{n}\sum_{i}^{n} V(\hat{y}_i,y_i)^2$$

Eq. 3. Root Mean Squared Error

$$R(\hat{Y},Y) = \sqrt{M(\hat{Y},Y)}$$

For quantitative measurements with variation, a corresponding prediction has a probability of occurring in real measurements. The prediction error should account for the probability of the prediction occurring. For two predictions corresponding to two different measurements with different measurement variations, the prediction that is more probable to occur within their corresponding probability distribution should score better than the prediction that is less probable, even when they both have the same absolute prediction error.

Notably, ML models can be used to predict the measurement variation by predicting the parameters of an assumed distribution, known as probabilistic learning (Murphy, 2022). For example, for measurement variation that is explained by a *t*-distribution, the model can predict the mean and sample standard deviation of the *t*-distribution. The prediction error, then, is determined by comparing the predicted and actual distributions. The objective of this study, however, is to assess a prediction that corresponds to the actual value.

The measurement variation in genomics data, which is the primary impetus of this study, is explored in **Section 2**. A novel adjustment to the prediction error that effectively accounts for measurement variation, called the **Deviation Error**, is introduced in **Section 3**.

Functions that are used to assess the predictions can often be used as loss functions (also known as cost functions): functions that are used to adjust the learnable parameters of a ML model. Typically, the loss function is the same as the function that is used to assess the predictions. Different loss functions have been used in previous studies to model genomics data. Towards a general prescription for modelling genomics data to reduce the Deviation Error, different loss functions are experimented on synthetic data that mimics genomics data in **Section 4**.

2 Measurement Variation in Genomics

Genomics is largely concerned with nucleotide sequences and the molecules that interact with them. For example, the strength of an interaction of a transcription factor to a specific region in the genome can vary. Measuring phenomena in genomics typically involves 4 steps: material preparation, sample preparation, DNA sequencing, and data processing.

Material preparation involves gathering biological material and fixing it. Fixing halts and preserves molecular interactions, creating a snapshot of the molecular state in the biological material.

Sample preparation involves isolating target nucleotide sequences (typically 25-250 nucleotides long) from the material prepared that is essentially a population of nucleotide sequences, referred to here as base-level nucleotide sequences. If the target nucleotide sequences

87 are RNA, they are converted into DNA sequences. The DNA sequences gathered from
88 experiments, referred to here as experimental DNA, are identified by DNA sequencing.
89 There are different methods for DNA sequencing, but High-Throughput Sequencing
90 (HTS) is the current standard and thus the focus of this study. The identified DNA sequences
91 from HTS are known as sequencing reads, simply referred to as reads hereafter.
92 Most often, the experimental DNA derives from known nucleotide sequences, referred to
93 here as reference sequences, which are typically the genome of the biological material. Data
94 processing entails mapping the reads to the reference sequences and assigning each position of
95 the reference sequences a score based on the number of reads mapping to it. The scores are
96 known as read counts and are non-negative integers.
97 DNA sequencing and sample preparation involve sampling steps which results in
98 measurement variation. HTS identifies a sample of the DNA in the experimental DNA.
99 Repeating HTS on the same experimental DNA can yield different reads. Then, for a particular
100 position of a reference sequence, the number of reads mapping to it can vary, referred to here as
101 sequencing variation. The sequencing variation of a read count is best explained by a Negative
102 Binomial distribution (NBD) (Love et al., 2014; McCarthy et al., 2012). NBDs differ depending
103 on the expected number of read counts, exemplified in **Fig. 1. A**. To estimate the expected
104 numbers of the read counts, sequencing must be repeated on the same experimental DNA,
105 referred to here as sequencing replicates.
106

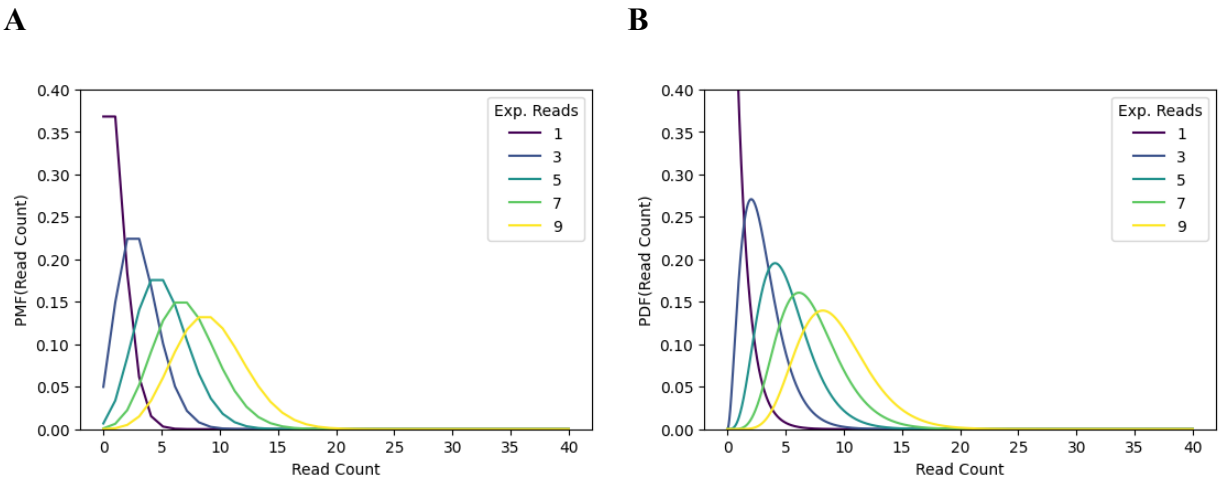


Fig. 1. The Gamma distribution is the continuous analog of the Negative Binomial distribution.

The Negative Binomial (**A**) and the Gamma (**B**) distributions with different expected read counts ("Exp. Reads").

The total number of reads produced from HTS can be roughly controlled. A greater number of reads yields better resolution than a lesser number of reads but costs more. Despite efforts to control the number of reads, the number of total reads for different sequencing replicates can vary. To make sequencing replicates comparable, the read counts of the sequencing replicates need to be scaled which is often based on the average total number of reads among the replicates.

Scaling often results in continuous quantities. Also, depending on the objective of the experiment and the number of reads used for HTS, the read counts may be smoothened by an averaging window, also yielding continuous values. The continuous analog of the NBD is the Gamma distribution (GD, **Sup. Eq. 2**), exemplified in **Fig. 1. B**. Notably, the GD requires the

actual value be greater than 0 while read counts can be 0. To address this, a constant that is the smallest computational accuracy is added to all read counts.

The read counts also contain two sources of bias, referred to here as sequence bias. First, some DNA sequences are more preferentially sequenced than others (Amemiya et al., 2019). Second, the number of base-level nucleotide sequences containing a particular position of the reference sequences can vary. To account for the sequence bias, a sample of base-level nucleotide sequences is sequenced, commonly known as an input control, referred to here as a sequencing control. The sequencing control also has sequencing variation which too requires replicates and is best modelled by a GD. The read counts of the target DNA are divided by the corresponding read counts of the sequencing control. The ratio of two GDs is a Beta Prime distribution (BPD) (**Sup. Eq. 3**, **Sup. Eq. 4**). The BPDs of different pairs of target and sequencing control read counts (simply target-control pairs hereafter) are exemplified in **Fig. 2**.

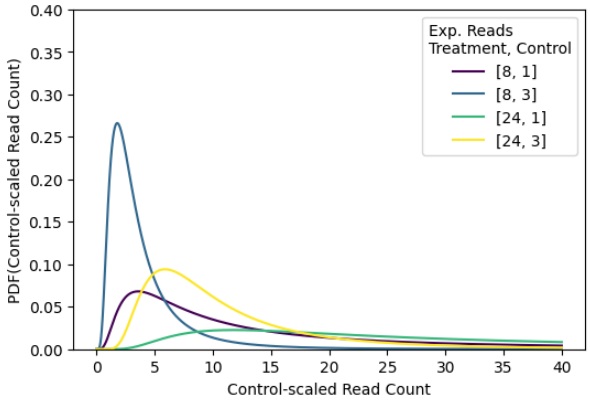


Fig. 2. Beta Prime distributions of different target-control pairs.

Target and control expected read counts ("Exp. Reads") are noted as [target, control].

139 Sample preparation typically entails numerous sampling steps, resulting in measurement
140 variation in read counts that too may be explained by NBDs. Still, sample preparation also
141 contains other variation that may not be adequately explained by NBDs. Estimating the variation
142 in sample preparation requires repeating sample preparation from the same biological material,
143 referred to here as sample preparation replicates. To distinguish the variation from sequencing
144 and from sample preparation, each sample preparation replicate needs to have sequencing
145 replicates.

146 Furthermore, phenomena in genomics can vary, referred to here as phenomenal variation.
147 For example, a transcription factor may interact with less variation to one location in the genome
148 than to another. To estimate the phenomenal variation, material preparation needs to be repeated,
149 known as biological replicates. To distinguish the phenomenal variation from measurement
150 variation (which includes both variation from sequencing and sample preparation), each
151 biological replicate needs to have sample preparation replicates.

152 Due to costs and/or limited biological material, most genomic experiments consist of 2-3
153 biological replicates of the target and 1-3 replicates of the sequencing control that may be
154 biological, sample preparation, or sequencing replicates depending on the experiment. With
155 relatively little information, the biological variation cannot be reliably distinguished from the
156 sequencing and sample preparation variation. So, the replicates for the experiment and the
157 sequencing control are treated as sequencing replicates. As mentioned, the variations of
158 measurements of control-scaled sequencing replicates are best explained by BPDs. Thus, for
159 genomics experiments with these or similar replicates, the variations of measurements are best
160 explained by BPDs.
161

3 Deviation Error

For measurements with measurement variation, the absolute prediction error does not account for measurement variation, exemplified for different target-control pairs in **Fig. 3**. As mentioned, the prediction error for measurements with measurement variation should account for measurement variation; it should account for the probability of observing the prediction in real measurements. This study assumes that predictions are for measurements with measurement variation.

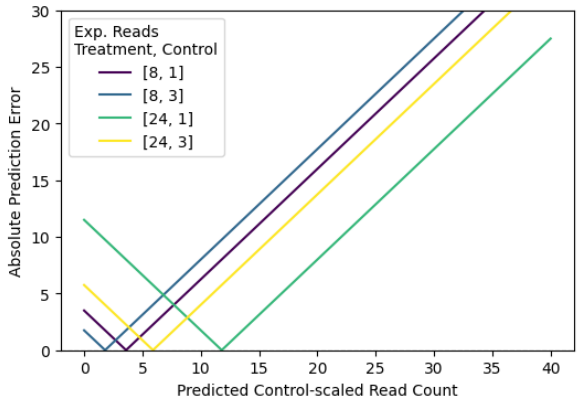


Fig. 3. Absolute prediction error does not account for measurement variation. See **Fig. 2** for additional descriptions.

Assessing predictions strictly based on their probability of occurring in real measurements yields values where greater values are preferred, known as reward functions. To assess a set of predictions of a model, a straightforward approach is to average the probabilities, referred to here as the Mean Probability (**Eq. 4**). One commonly used approach to assess a set of predictions is the product of the probabilities since each prediction corresponds to an independent observation, known as the likelihood (**Eq. 5**). To avoid arithmetic underflow, the

Log of the Likelihood is used (**Eq. 6**) which is the sum of the log of the probabilities. To avoid log(0) which is undefined, a constant is added. Notably, the distribution of values in log scale are different than the distribution of values in normal scale and the distribution depends on the constant added.

Eq. 4. Mean Probability

$$Q(\hat{y};M) = \frac{1}{n}\sum_{i}^{n} P(\hat{y}_i; m_i)$$

where:

$m =$ an arbitrary set of function parameters, $m \in M$

$P(x;m) =$ probability distribution function

Eq. 5. Likelihood

$$L\left(\theta|\hat{Y},M\right) = \prod_{i=1}^{n} P(\hat{y}_i; m_i)$$

Eq. 6. Log Likelihood

$$\log L\left(\theta|\hat{Y},M\right) = \sum_{i=1}^{n} \log(P(\hat{y}_i) + q;m_i)$$

where:

$q =$ a constant to avoid log(0)

$\theta =$ model parameters

Assessing predictions by metrics where lower values are preferred, known as error functions, are more useful as they can be used as loss functions. A straightforward approach to convert reward functions to error functions is to use the negative (i.e. the reflection) of the reward function. The negative of the Mean Probability, referred to here as the Negative Mean Probability (NPM, **Eq. 7**) can be used. The negative of the Log Likelihood, known as the Negative Log Likelihood NLL, **Eq. 8**) is a commonly used function in ML and statistics.

Eq. 7. Negative Mean probability

$$J(\hat{Y};M) = -Q(\hat{Y};M)$$

Eq. 8. Negative Log Likelihood

$$K(\theta|\hat{Y},M) = -\log L(\theta|\hat{Y},M)$$

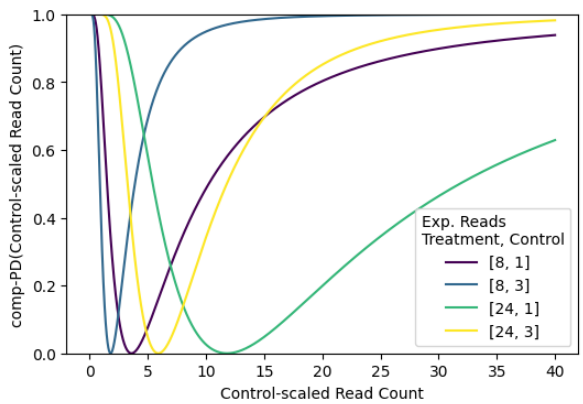
Importantly, different probability distributions can have different maximums, exemplified previously in **Fig. 2**. Accurately predicting the mode of one measurement variation may not reward/penalize the same as predicting the mode of another. Following the example in **Fig. 2**, predicting the mode of a target-control pair of [8,3] yields a greater reward or less error than predicting the mode of a target-control pair of [8,1].

To comparably assess predictions corresponding to different probability distributions, the probability distributions can be scaled by their respective maximums, referred to here as maximum-scaled probability distributions (ms-PDs). Notably, maximum-scaled PDFs describe the probability of observing a continuous value within an infinitesimally small interval relative to observing the mode within an equally infinitesimally small interval. Also notably, the values of

223 an ms-PD have a range of [0, 1]. To make an error function where lower values are preferred and
224 not needing to use log which alters the shape of the distribution, the complement of the ms-PD
225 can be used, referred to here as the as the complement probability distribution (comp-PD) (**Eq.**
226 **9**). Following the example in **Fig. 2**, the comp-PDs of different target-control pairs are
227 exemplified in **Fig. 4**.
228
229 **Eq. 9. Complement Probability Distribution**

230 $$C(x;m) = 1 - \frac{P(x;m)}{\max(P(x;m))}$$

231



232
233 **Fig. 4. Comp-PDs of different target-control pairs.**
234 See **Fig. 2** for additional descriptions.
235 The modes of the BPDs are determined using **Sup. Eq. 5**.
236
237 An issue with using the ms-PD or the comp-PD for assessing predictions is that
238 probability distributions plateau as predictions are increasingly away from the mode: ms-PDs
239 approach 0 and comp-PDs approach 1. Following the example in **Fig. 4**, predictions that are
240 greater than 20 for a target-control pair of [8,3] are all similar and close to 1. To account for both

the probability of observing a prediction in real measurements and the absolute prediction error, the **Deviation Error** is developed. The Deviation Error is the product of the prediction error and the comp-PD (**Eq. 10**). Following the example in **Fig. 2**, the Deviation Error for different target-control pairs is exemplified in **Fig. 5**. The Deviation Error can directly substitute for the absolute prediction error in larger functions that use it. For example, the Deviation Error can be used in the MSE, called the Mean Squared Deviation Error (MSDE, **Eq. 11**), or the RMSE, called the Root Mean Squared Deviation Error (RMSDE, **Eq. 12**).

Eq. 10. Deviation Error

$$D(\hat{Y},Y) = (V(\hat{Y},Y)) * (C(\hat{Y};M))$$

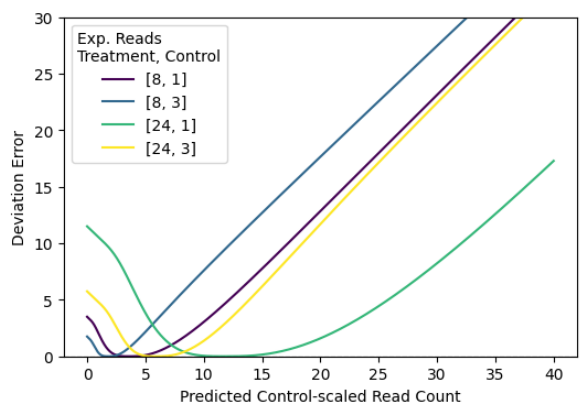


Fig. 5. Deviation error accounts for prediction error and probability.

See **Fig. 2** for additional descriptions.

Eq. 11. Mean Squared Deviation Error

$$M_D(\hat{Y},Y) = \frac{1}{n}\sum_{i}^{n} D(\hat{y}_i,y_i)^2$$

Eq. 12. Root Mean Squared Deviation Error

$$R_D(\hat{Y},Y) = \sqrt{M_D(\hat{Y},Y)}$$

4 Synthetic Genomics Data

The objective in modelling measurements with measurement variation is to produce predictions that have minimal Deviation Error. Intuitively, to minimize the Deviation Error of the predictions, model fitting should use the Deviation Error in the loss function. Still, the effects on performance of using the Deviation Error in the loss function compared to other loss functions is not evident. The following case study compares using the Deviation Error in the loss function to other loss functions towards a general prescription for modelling measurements with measurement variation. More specifically, the following case study aims to provide a general prescription for reducing the RMSDE on modelling genomics data.

Genomics data mostly involves sequences which are best modelled by Neural Networks (NNs). There have been several different loss functions that have been used for fitting NNs on genomics data. The simplest loss functions used outside in ML in general include the MSE (e.g. Alipanahi et al., 2015) and its closely related RMSE. Another commonly used strategy in ML is to model the standardized measurements: measurements converted into *z*-scores. Using the MSE on the *z*-scores (e.g. Koo et al., 2021) is referred to here as Mean Squared *z*-score Error (MSZE, **Eq. 13**). The methods mentioned thus far do not consider measurement variation.

Eq. 13. Mean Squared *z*-score Error

$$M_z(\hat{Y},Y,\mu_Y,\sigma_Y) = \frac{1}{n}\sum_i^n \left(\frac{\hat{y}-\mu_Y}{\sigma_Y} - \frac{y-\mu_Y}{\sigma_Y}\right)^2 = \frac{1}{n}\sum_i^n \left(\frac{\hat{y}-y}{\sigma_Y}\right)^2$$

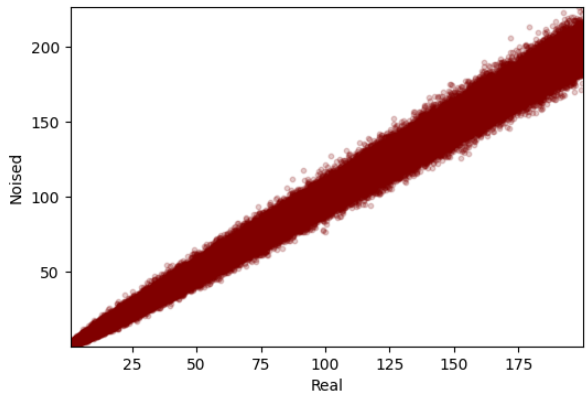
where:

$\mu_Y =$ mean of Y

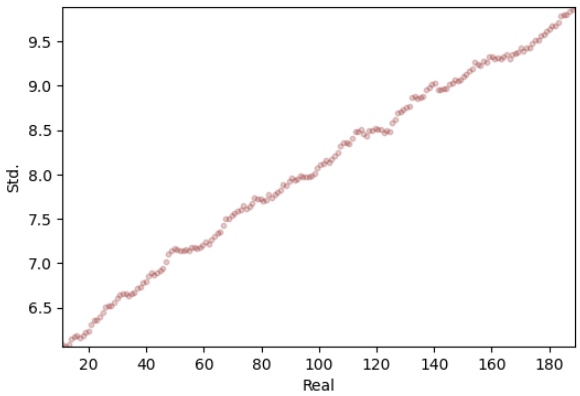
$\sigma_Y =$ standard deviation of Y

As mentioned, the variation of a measurement in genomics is often best explained by a GD or a BPD for experiments without or with sequencing controls, respectively. Both distributions exhibit variation that increases with the magnitude of the values (exemplified in **Fig. 6**), resembling multiplicative variation. A common strategy to model measurements with multiplicative variation is to model in log scale as it converts multiplicative relationships to additive relationships which is preferred (see **Sup. Note 9.1.2**). Using the MSE on log scaled values is known as Mean Squared Log Error (MSLE, **Eq. 14**). One issue is that genomics measurements can be 0 but log(0) is undefined, thereby requiring the addition of a constant. A constant of 1 is typically used (e.g. Avsec et al., 2021), referred to here as MSLE-1. Another similar approach that accounts for multiplicative variation is the Mean Poisson Negative Log Likelihood (P-NLL, **Eq. 15**) (e.g. Kelley, 2020).

A



B



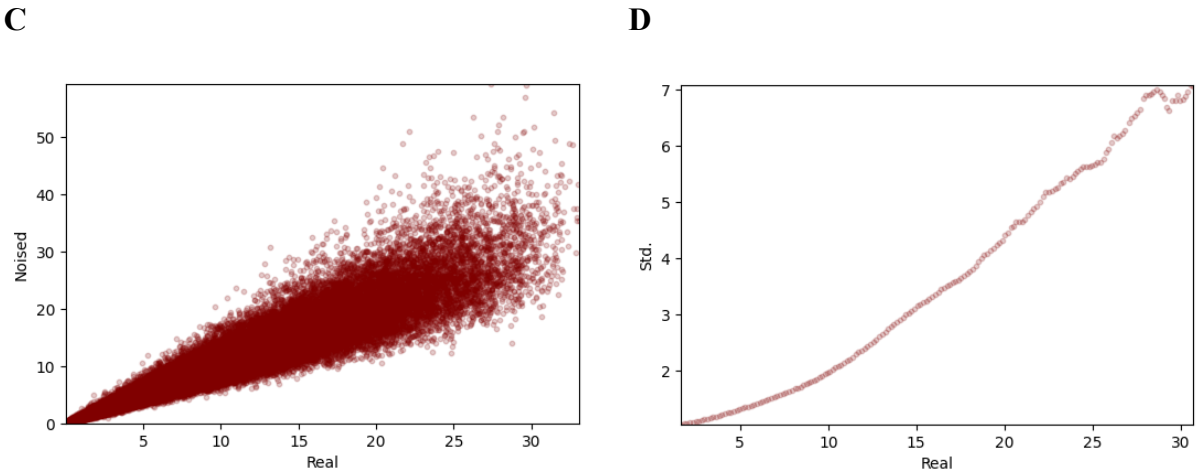


Fig. 6. Measurement variation explained by GDs and BPDs resemble multiplicative variation.

Mimicking genomics measurements, 100,000 target-control pairs were generated. The *real* target values were drawn from a uniform distribution with a minimum of 1 and a maximum of 200. The real control values were drawn from a uniform distribution with a minimum of 5 and a maximum of 20. The modes of the real target-control pairs for BPDs were determined. For each of the real target and real control values, 3 samples were drawn from a GD with an alpha of the real value and a scale of 1 and averaged. The modes of sampled target-control pairs for BPDs were determined.

A, C) Scatter plots compare the real and the noised values for the target (**A**) which follow GDs and for the target-control pairs which follow BPDs (**C**).

B, D) Scatter plots comparing the mean and standard deviations of sliding windows with a size 0.1x and stride of 0.005x the total range of the Noised values in **A** (**B**) and in **C** (**D**).

The windowed standard deviations of the measurements with measurement variation that is explained by GDs and BPDs increase as the real value increases.

Eq. 14. Mean Squared Log Error

$$M_L(\hat{Y},Y,q) = \frac{1}{n}\sum_{i=1}^{n}(\log(\hat{y}_i + q) - \log(y_i + q))^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\log\left(\frac{\hat{y}_i + q}{y_i + q}\right)\right)^2$$

Eq. 15. Mean Poisson Negative Log Likelihood

$$U(\hat{Y},Y,q) = \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i - y_i * (\log(\hat{y}_i + q))$$

As explored, the NMP, NLL, and the Deviation Error can all be used to assess predictions for measurements with measurement variation. The NMP and the NLL require determining the probabilities of a prediction occurring. For continuous measurements that use a PDF, determining the probability of observing a value within an interval requires integration of the PDF which is facilitated by using the Cumulative Density Function (CDF) of the PDF. Implementing the CDF of a GD or a BPD in a loss function is currently non-trivial due to software limitations. As mentioned, a maximum-scaled PDF describes the probability of observing a continuous value relative to observing the mode of its probability distribution. For this case study, the NMP and the NLL are tried as loss functions using maximum-scaled probabilities, referred to here as ms-NMP and ms-NLL, respectively. Conversely, the Deviation Error already uses maximum-scaled probability distributions. The RMSDE can be directly used as a loss function and thus is also tried.

A NN is used to generate data consisting of target-control pairs that simulates real genomics experiments. Different datasets varying in the number of observations and additional noise are generated to simulate scenarios with different amounts of data and noise from other sources. A similar NN is used to model the data.

4.1 Methods

All code is available at https://github.com/Husam94/DevLoss.

4.1.1 Neural Networks

Inspired by Steininger et al., 2021, all NNs used have 3 Dense layers each with 10 nodes and ReLU activation functions between the Dense layers. Batch normalization layers were added between the Dense layers as they are found in most of the current NN architectures. For data generation, a NN with two outputs is used to produce target-control pairs, referred to here as the Data NN. For modelling, a NN with one output is used to predict the modes of the BPDs of the target-control pairs, referred to here as the Model NN. The Model NN has a ReLU activation function at the end so that values have a minimum of 0.

4.1.2 Data Generation

4.1.2.1 Initial Data Generation

A 10-feature input was generated by sampling uniform distributions with a range [-1, 1]. A total of 1,000,000 10-feature inputs were generated. A Data NN was randomly initialized by Kaiming Initialization and produced corresponding 2-feature outputs from the inputs. The pairs of input-output are referred to as original data. This procedure was repeated 20 times, generating 20 sets of original data from 20 different initializations of Data NNs, referred to as synthetic repeats. The outputs of the synthetic repeats were visually inspected for bell-shaped distributions.

Of the synthetic repeats, 5 were chosen that best resemble bell-shaped distributions for the target and control values.

4.1.2.2 Noise

The following applies to each of the synthetic repeats. Different *levels* of noise – 0, 0.5, 1.0 referred to as N1, N2, N3, respectively – were added to the outputs of the original data. The noise is the product of the level of noise, the standard deviation of the original outputs, and samples from a truncated normal distribution with a mean of 0, a standard deviation of 1, and bounds of ±2 standard deviations (**Eq. 16**), exemplified in **Fig. 7**. Notably, the noise level N1 is devoid of noise.

Eq. 16.

$$y_{noised} = y_{original} + \left(\sigma^2_{original} * g * \mathrm{TN}(0,1,2)\right)$$

where:

$y_{original} =$ output values of an original data

$\sigma^2_{original} =$ standard deviation of the output values of an original data

$g =$ level of noise

$\mathrm{TN}(\mu,\sigma^2, b) =$ truncated normal distribution with a mean (μ), standard deviation (σ^2) and bounds of $\pm b$

$y_{noised} =$ output values of an original data with noise

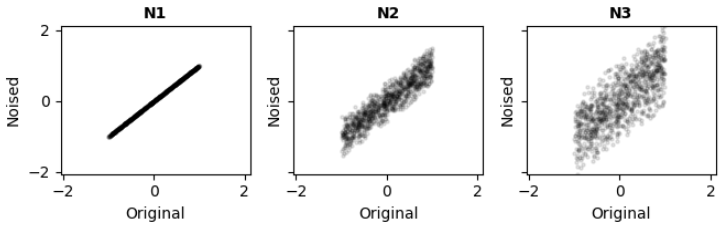


Fig. 7. Examples of levels of noise.

Different levels of noise were added ("Noised") to a random sample of 1,000 original outputs ("Original").

4.1.2.3 Selection

The following applies to each level of noise. After adding noise, observations with targets and controls within ±1 standard deviation of their respective distributions were kept for selection. It was preferred that the observations have a diverse combination of target-control values. The target and control values were separately binned to 10 equally spaced bins. The observations were binned as a combination of their target and control bins, producing (10*10) 100 bins.

Different numbers of observations – 5000, 10000, and 20000, referred to as O1, O2, O3, respectively – were desired. The observations were iteratively selected without replacement from each of the 100 bins (in random order) until the desired number of observations were obtained.

4.1.2.4 Sampling from Gamma Distributions

The following applies to each selected dataset. Based on personal experience with real genomics data, the target and the control values were scaled to ranges [0, 200] and [5, 20], respectively, referred to as real values. For each of the real target and control values, 3 samples were drawn from respective GDs with alphas of the real values and a scale of 1, mimicking

398 sequencing replicates. The samples were then averaged, referred to as actual values. The modes
399 of the actual target-control pairs for BPDs were determined.
400
401 **4.1.2.5 Data Splitting**
402 For each dataset, the observations were binned into 10 equally spaced bins by the mode
403 of the actual target-control pairs. The data was split into training, stoppage, and testing sets with
404 proportions of 0.6, 0.2, 0.2, respectively, by Stratified split (see **Suppl. Note 9.1.1**)
405
406 ### 4.1.3 Scoring Experimental Units
407 The objective is to reduce the RMSDE of the modes of the actual target-control pairs.
408 The experimental units are assessed by the performance on the testing set measured by the
409 RMSDE.
410 The learnable parameters of NNs are initialized randomly and are fit using stochastic
411 methods which can result in substantially varying model performances. Considering computing
412 budgets, each experimental unit was repeated 15 times, referred to as model repeats. Random
413 initializations can result in models that produce predictions that are only below 0 before the final
414 ReLU activation and thus produce predictions that are all 0 after ReLU activation which disables
415 model learning, referred to as a dud initialization. To avoid dud initializations, a model with a
416 dud initialization is re-initialized up to 15 times. For later statistical assessment, different sets of
417 model repeats were simulated by bootstrapping the existing model repeats (Efron & Tibshirani,
418 1986). Based on computing budgets, 100 bootstrap sets of model repeats were sampled for each
419 set of model repeats.

Commonly, a single model out of a set of model repeats that best performs on the stoppage set is used for further assessment. This selection criteria, however, lacks reproducibility as a different set of model repeats can yield a substantially different best performing model. Instead of using a single best performing model, the best 3 models were used. The predictions of the best 3 models were averaged, akin to an average ensemble model, simply referred to as an ensemble. The ensemble predictions were scored by the RMSDE, referred to as a bootstrap score. Each experimental unit had 100 bootstrap scores as there were 100 bootstrap sets of model repeats.

4.1.3.1 Comparisons of Experimental Units

A subject experimental unit was compared to a corresponding reference experimental unit by pairwise comparisons of their bootstrap scores, resulting in (100*100) 10,000 comparisons which simulates repeating the modelling of the two experimental units 10,000 times. Each pair of bootstrap scores were compared by percent relative change, referred to as the pairwise bootstrap score. Negative pairwise bootstrap scores indicate the subject has a lower RMSDE than the reference which is preferred. For each number of observations, for each level of noise, the pairwise bootstrap scores of synthetic repeats were pooled, totaling (5*10,000) 50,000 pairwise bootstrap scores.

The mean and the standard error of the pooled pairwise bootstrap scores were determined for each dataset and for each experimental unit. For statistical assessment, the 5th (P_5) and the 95th (P_{95}) percentiles of the pairwise bootstrap scores were determined, simulating a one-sided statistical significance level of 0.05 and a two-sided statistical significance level of 0.1 (Efron & Tibshirani, 1986). If $0 > P_{95}$, percentile, the subject model is significantly better than the

reference model. If $0 < P_5$ percentile, the subject model is significantly worse than the reference model. If $P_5 < 0 < P_{95}$, the subject model is not significantly different from the reference model.

4.1.4 Model Fitting

All modelling used the Model NN to predict the mode of the actual target-control pairs, a batch size of 512, a stoppage patience of 10 epochs (see **Suppl. Note 9.1.1**), and the Adam Optimizer with a learning rate of 0.0124 (Kingma & Ba, 2014). The learning rate used for the experiments was determined as the learning rate that best reduces the RMSDE on the validation set with MSE as the loss function and a budget of 200 epochs (**Sup. Table. 1**).

Unless otherwise specified, a constant value of 1e-10 is used for all loss functions to avoid log(0) or dividing by 0.

4.2 Results and Discussion

Models were fit with different loss functions on different sets of data that varied by number of observations and levels of noise. For each set of data, the sets of models fit with different loss functions were compared to models fit using MSE as the loss function (**Table 1**). Generally, similar results were produced with different numbers of observations as well as for noise levels N2 and N3. For simplicity, only the results for datasets with a number of observations of O2 and with noise levels of N1 and N3 are further analyzed.

Table 1. Compared to using MSE as the loss function, using any other loss function does not significantly improve model performance.

Mean pairwise bootstrap scores comparing fitting with different loss functions to fitting with RMSE as the loss function. See **Sup. Table. 2** for full bootstrap statistics.

Negative values indicate that models fit using the subject configuration performed better than models fit using the reference configuration. Bolded values indicate that the models fit using the subject configuration are significantly better or worse than models fit using the reference configuration.

"Num. Obs. ID" refers to the number of observations ID (e.g. O1 is 5000 observations). Datasets that are focused on in further analysis are bordered with thicker lines. Using the Deviation Error in a loss function (RMSDE) is shaded in yellow.

Num. Obs. ID	**O1**			**O2**			**O3**		
Noise ID	**N1**	**N2**	**N3**	**N1**	**N2**	**N3**	**N1**	**N2**	**N3**
Loss Functions									
MSE	0.2	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0
RMSE	-6.1	0.5	-0.1	-0.2	0.4	0.1	2.1	-0.2	0.0
MSZE	-3.1	1.1	-0.3	-1.0	-0.3	0.2	1.9	0.5	0.1
MSLE-1	12.1	**5.5**	**7.4**	7.6	**7.2**	**7.8**	-2.5	**7.7**	**8.0**
P-NLL	3.2	-1.3	-0.9	4.4	0.2	-0.2	3.3	0.5	-0.1
ms-NMP	0.4	**5.3**	**2.8**	7.8	**3.2**	**3.1**	2.2	**3.6**	**1.8**
ms-NLL	**15.2**	2.7	**1.8**	**23.5**	**3.4**	**2.2**	9.3	**3.3**	**1.9**
RMSDE	-0.7	0.8	0.1	2.3	0.0	-0.1	4.7	0.9	-0.1

Compared to model fitting with MSE as the loss function, model fitting with any other loss function produced models that were not significantly better. Model fitting with RMSE, MSZE, P-NLL, or RMSDE as the loss function produced models that were not significantly better or worse: they produced models that performed similarly to those that were fit with MSE as the loss function for both N1 and N3. Conversely, model fitting with MSLE-1, ms-NMP, or ms-NLL as the loss function produced models that were significantly worse on datasets with noise levels of N3.

As explained, the MSLE-1 uses log scaled values to assess predictions for measurement variation that is multiplicative. Adding a constant prior to log scaling disrupts the conversion of multiplicative measurement variation to additive measurement variation (see **Sup. Note 9.1.2)**. This holds true for measurement variations that can be explained by BPDs and a constant of 1, exemplified in **Fig. 8**. Still, the performance losses from fitting with MSLE functions are presumably attributed to comparison of the prediction to the actual as a ratio (**Eq. 14**) which is relative prediction error while the RMSDE considers absolute prediction error. For predictions with the same absolute prediction error but corresponding to different actuals, the relative absolute error is more pronounced for the prediction corresponding to the smaller actual. This influences model fitting to prioritize predictions for lower values. The added constant also affects the quotient. In experiments not included here, different constants result in different performances. Notably, using MSLE-1 as a loss function for datasets with noise level N1 – which is devoid of any noise – produced models that were not statistically different than using MSE.

A **B**

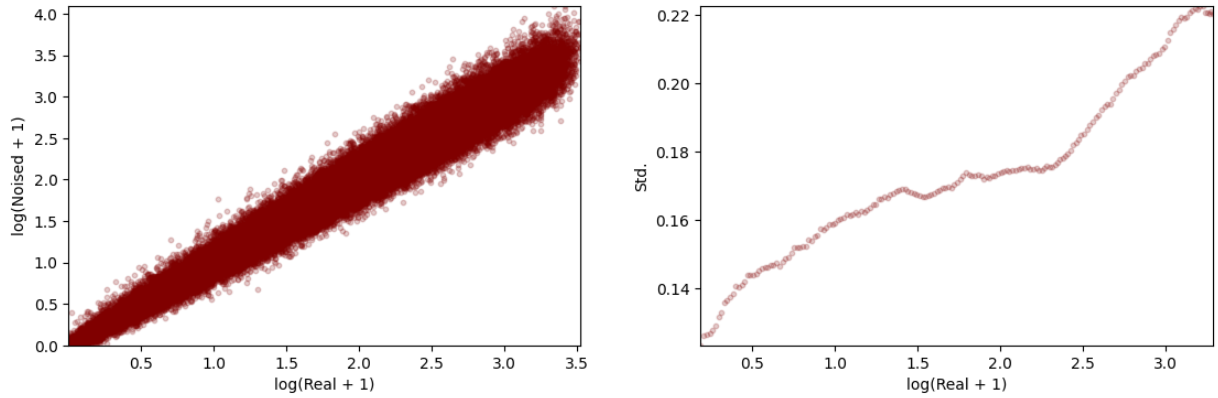


Fig. 8. Adding a constant of 1 prior to log scaling does not convert beta-prime variation into additive variation.

A) Following the example in **Fig. 6 C**, a constant of 1 is added to real and noised values prior to log scaling.

B) Scatter plots comparing the mean and standard deviations of sliding windows with a size 0.1x and stride of 0.005x the total range of the Noised values in **A**.

The ms-NMP and the ms-NLL are solely based on the on the probability of observing the prediction in real measurements. The Deviation Error, however, assesses the probability of observing the prediction as well as the absolute prediction error. The performance differences are presumed to be from not considering the absolute prediction error. The performance differences of the ms-NMP and the ms-NLL are presumably attributed to the use of log in the ms-NLL. Notably, using ms-NMP as a loss function for datasets with noise level N1 – which is devoid of any noise – produced models that were not statistically different than using MSE.

Using the Deviation Error in a loss function (RMSDE) did not yield models that performed significantly different than using loss functions that did not use the Deviation Error even as the models were assessed by the Deviation Error. Using the Deviation Error as a loss function may require a different learning rate for optimal performance than the one used in this experiment which was optimized for using MSE as the loss function. Still, the same can be said for all the other loss functions except for MSE.

It is recommended that either MSE, RMSE, P-NLL, or RMSDE be used as loss functions when the objective is to produce models that reduce the RMSDE regardless of noise level. Similar conclusions are expected for reducing the MSDE as it is closely related to the RMSDE.

5 Conclusion

The Deviation Error is a novel metric that effectively accounts for both absolute prediction error and measurement variation. It is recommended to be used for assessing predictions corresponding to measurements with measurement variation. On synthetically generated data that mimics genomics data, using the Deviation Error in the loss function for fitting NNs yielded models that performed similarly to those that were fit using the MSE as the loss function.

6 Acknowledgements

None.

7 Author Contributions

Conceptualization: Husam Abdulnabi

Data curation: Husam Abdulnabi

Formal analysis: Husam Abdulnabi

Investigation: Husam Abdulnabi

Methodology: Husam Abdulnabi

Project administration: J. Timothy Westwood

Software: Husam Abdulnabi

Supervision: J. Timothy Westwood

Writing – original draft: Husam Abdulnabi

Writing – review & editing: J. Timothy Westwood

8 References

Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, *33*(8), 831–838. https://doi.org/10.1038/nbt.3300

Amemiya, H. M., Kundaje, A., & Boyle, A. P. (2019). The ENCODE Blacklist: Identification of Problematic Regions of the Genome. *Scientific Reports*, *9*(1), 9354. https://doi.org/10.1038/s41598-019-45839-z

Avsec, Ž., Weilert, M., Shrikumar, A., Krueger, S., Alexandari, A., Dalal, K., Fropf, R., McAnany, C., Gagneur, J., Kundaje, A., & Zeitlinger, J. (2021). Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, *53*(3), 354–366. https://doi.org/10.1038/s41588-021-00782-6

Efron, B., & Tibshirani, R. (1986). Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy. *Statistical Science*, *1*(1), 54–57.

Kelley, D. R. (2020). Cross-species regulatory sequence activity prediction. *PLoS Computational Biology*, *16*(7), e1008050. https://doi.org/10.1371/journal.pcbi.1008050

Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*.

Koo, P. K., Majdandzic, A., Ploenzke, M., Anand, P., & Paul, S. B. (2021). Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLoS Computational Biology*, *17*(5), e1008925. https://doi.org/10.1371/journal.pcbi.1008925

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*(12), 550. https://doi.org/10.1186/s13059-014-0550-8

McCarthy, D. J., Chen, Y., & Smyth, G. K. (2012). Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Research*, *40*(10), 4288–4297. https://doi.org/10.1093/nar/gks042

Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT Press.

Steininger, M., Kobs, K., Davidson, P., Krause, A., & Hotho, A. (2021). Density-based weighting for imbalanced regression. *Machine Learning*, *110*(8), 2187–2211. https://doi.org/10.1007/s10994-021-06023-5

9 Supplementary

9.1 Notes

9.1.1 Model Fitting

NNs are fit using data in the training set and in cycles called epochs. The number of epochs used in fitting can be set prior to modelling. However, more commonly, fitting continues until the model's performance on data in a set other than the training and testing sets – commonly known as the validation set but is referred to here as the stoppage set – deteriorates. An extra number of epochs is allotted to allow the model to regain performance, known as patience. The performance on the stoppage set is assessed by a function known as the stop criteria which is the loss function in this study.

9.1.2 Additive and Multiplicative Measurement Variation

Additive measurement variation is measurement variation that is (or at least assumed to be) constant for all measurements (**Eq. 17**), exemplified in **Fig. 9**. As a constant, additive measurement variation can be, and often is, safely ignored for assessment.

Eq. 17. Additive Measurement Variation

$$y = t + a$$

where:

$t =$ actual value

$a =$ randomly drawn from an arbitrary probability distribution $P(x;m)$

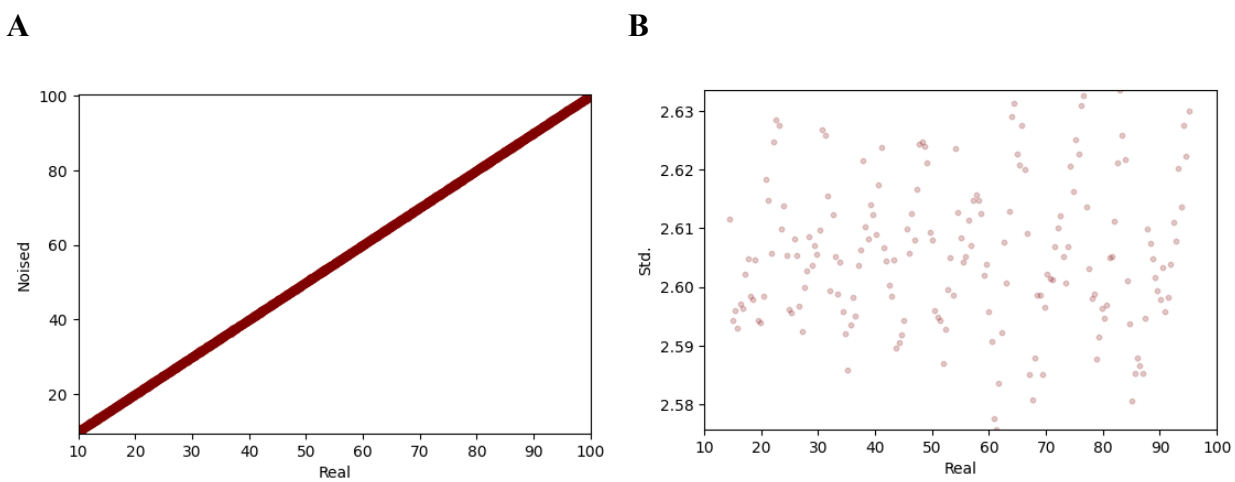


Fig. 9. Additive measurement variation.

Real values (100,000 points) were drawn from a uniform distribution with a minimum 10 and a maximum 100. Noise for each actual value was drawn from a normal distribution with a mean 0 and a standard deviation 0.2.

A. Noise was applied additively producing "Noised" values.

B. The mean and standard deviations of sliding windows with a size 0.1x and stride of 0.005x the total range of the Noised values in **A**.

The windowed standard deviations of the additively applied noise are relatively consistent for all values.

Multiplicative measurement variation is measurement variation that is (or at least assumed to be) directly proportional to the actual values (**Eq. 18**, exemplified in **Fig. 10 A, B**). Measurements with multiplicative variation can be modelled in log scale to convert multiplicative variation to additive variation (**Eq. 19**), exemplified in **Fig. 10 B, C**. Still, the predictions in log scale should be converted back to normal scale for assessment.

Eq. 18. Multiplicative Measurement Variation

$$y = t + t * a = t * (1 + a)$$

Eq. 19. Multiplicative to Additive Measurement Variation

$$\log(y) = \log(t * (1 + a)) = \log(t) + \log(1 + a)$$

A **B**

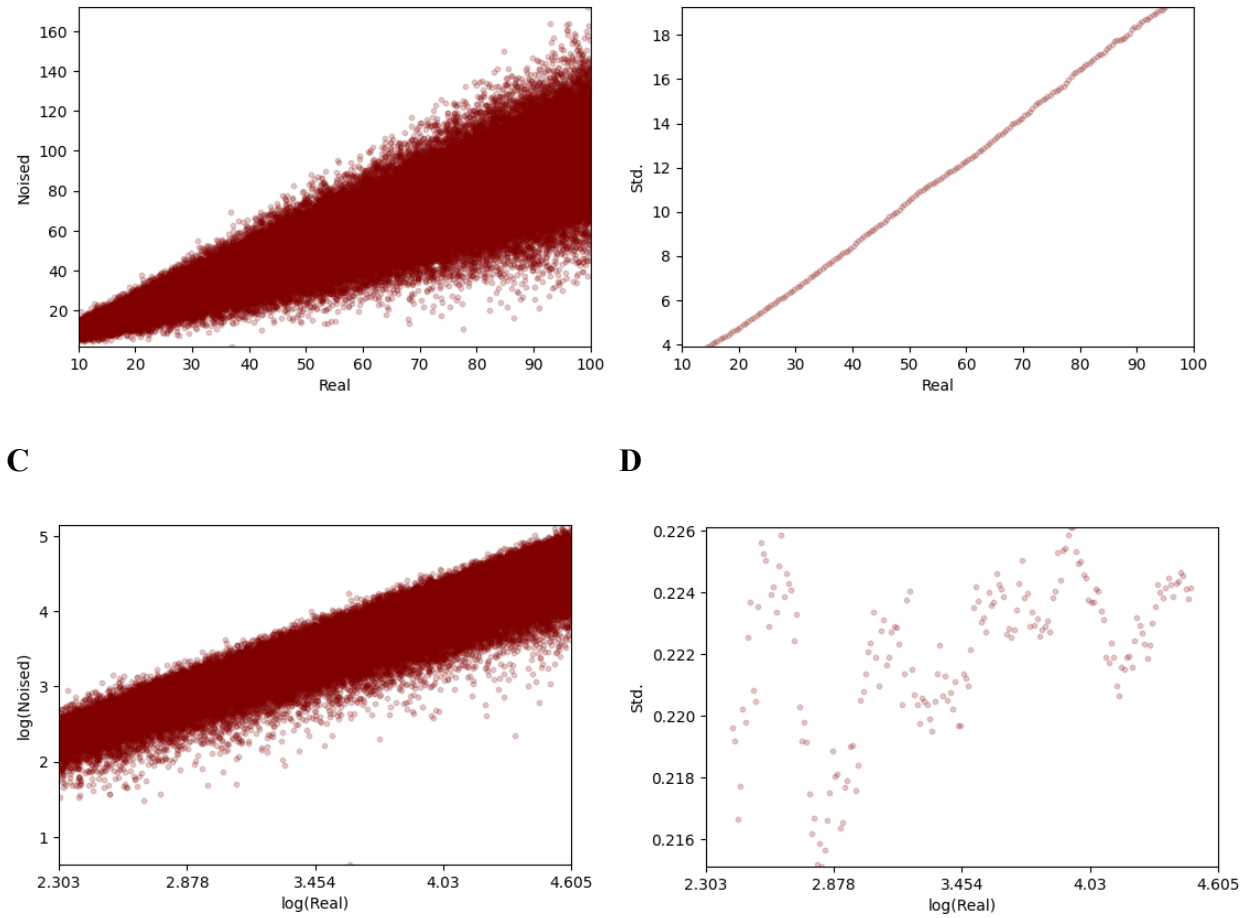


Fig. 10. Multiplicative measurement variation is additive measurement variation in log scale.

Following the example in **Fig. 9**, noise was applied multiplicatively instead of additively (**A, B**). The values were converted into log scale (**C, D**). The windowed standard deviations of the multiplicatively applied noise increase linearly. Conversely, the windowed standard deviations of the log scaled multiplicatively applied noise are relatively consistent for all values, resembling additive measurement variation in **Fig. 9**.

Importantly, log-scaling values requires that all values are greater than 0. For measurements that can be equal or less than 0, a constant must be added prior to log scaling. However, this disrupts the multiplicative to additive conversion, exemplified in **Fig. 11**.

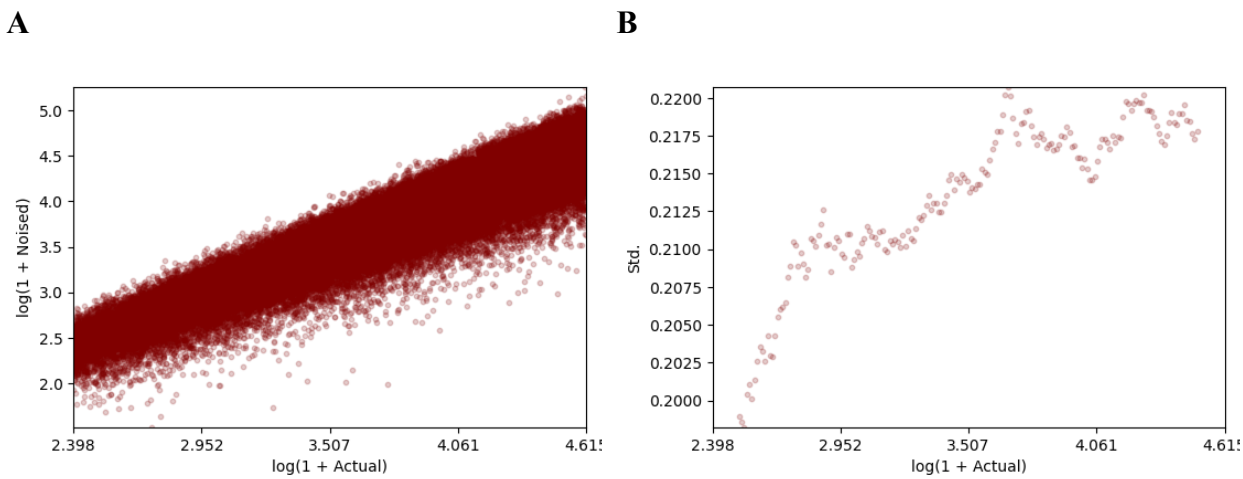


Fig. 11. Adding a constant prior to log scaling disrupts the multiplicative to additive noise conversion.

Following the example in **Fig. 10**, a constant of 1 is added to the multiplicatively applied noise (**Fig. 10 C**) before log scaling (**A**) and determining the mean and standard deviations of sliding windows (**B**).

9.2 Equations

Sup. Eq. 1. Relative Probability of a Probability Density Function

$$\frac{\Pr(y_1 - \varepsilon \leq y_1 \leq y_1 + \varepsilon)}{\Pr(y_2 - \varepsilon \leq y_2 \leq y_2 + \varepsilon)} = \frac{P(y_1)}{P(y_2)}$$

where:

$\varepsilon =$ a constant

Sup. Eq. 2. Gamma Distribution

$$G(x;\alpha,\theta) = \frac{(x^{\alpha-1})\left(e^{-\frac{x}{\theta}}\right)}{\Gamma(\alpha)\,(\theta^{\alpha})}, \alpha > 0, \theta > 0$$

where:

$\Gamma(x) =$ gamma function

$\alpha =$ "shape" parameter

$\theta =$ "scale" parameter

Sup. Eq. 3. Beta Prime distribution

$$H(x;\alpha,\beta) = \frac{(x^{\alpha-1})\left((1-x)^{\beta-1}\right)}{\mathrm{B}(\alpha,\beta)}, \alpha > 0, \beta > 0$$

where:

$$\mathrm{B}(\alpha,\beta) = \frac{(\Gamma(\alpha))\,(\Gamma(\beta))}{\Gamma(\alpha+\beta)}$$

Sup. Eq. 4. Ratio of two Gamma variables is a Beta Prime variable

$$\frac{G(x;\alpha_1,\theta_1)}{G(x;\alpha_2,\theta_2)} = \frac{\theta_1}{\theta_2} H(x,\alpha_1,\alpha_2)$$

Sup. Eq. 5. Mode of Beta Prime distribution

$$\mathrm{mode}(H(x,\alpha,\beta)) = \begin{cases} \dfrac{\alpha-1}{\beta+1}, & \alpha \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad \alpha > 0, \beta > 0$$

9.3 Tables

All tables in **DevLoss.Sup.Tables.xlsx.**

Sup. Table. 1. Mean pairwise bootstrap scores comparing fitting with Adam learning rates from 0.0001 to 0.05 to models fit with an Adam learning rate of 0.001.
Models were fit with MSE as the loss function.
Negative values indicate that models fit using the subject configuration performed better than models fit using the reference configuration. Bolded values indicate that the models fit using the subject configuration are significantly better or worse than models fit using the reference configuration, respectively.

Sup. Table. 2. Statistics of pairwise bootstrap scores comparing fitting with different loss functions to fitting with RMSE as the loss function.
"SE", "Conf. Int.", and "Sig." refer to standard error, confidence interval, and significance, respectively. Bolded rows indicate rows that are significant.