RESEARCH

MAVE-NN: Quantitative Modeling of Genotype-Phenotype Maps as Information Bottlenecks

Ammar Tareen[1], William T. Ireland[2], Anna Posfai[1], David M. McCandlish[1] and Justin B. Kinney[1*]

Abstract

Multiplex assays of variant effect (MAVEs) are being rapidly adopted in many areas of biology including gene regulation, protein science, and evolution. However, inferring quantitative models of genotype-phenotype maps from MAVE data remains a challenge. Here we introduce MAVE-NN, a neural-network-based Python package that addresses this problem by conceptualizing genotype-phenotype maps as information bottlenecks. We demonstrate the versatility, performance, and speed of MAVE-NN on a diverse range of published MAVE datasets. MAVE-NN is easy to install and is thoroughly documented at https://mavenn.readthedocs.io.

Keywords: MAVE; Neural Networks; Noise Agnostic Regression; Global Epistasis Regression

Background

Over the last decade, the ability to quantitatively study genotype-phenotype (G-P) maps has been revolutionized by the development of multiplex assays of variant effect (MAVEs), which can measure molecular phenotypes for thousands to millions of genotypic variants in parallel [1]. MAVE is an umbrella term that describes a diverse set of experimental methods [2, 3], three examples of which are illustrated in Fig. 1. Deep mutational scanning (DMS) experiments are one large class of MAVE [4]. These work by linking proteins [5, 6, 7] or structural RNAs [8, 9, 10, 11] to their coding sequences, either directly or indirectly, then using deep sequencing to assay which variants survive a process of activity-dependent selection (Fig. 1a). Massively parallel reporter assays (MPRAs) are another major class of MAVE [12, 13, 14, 15], and are commonly used to study DNA or RNA sequences that regulate gene expression at a variety of steps, including transcription [16, 17, 18, 19, 20, 21], splicing [22, 23, 24, 25, 26], polyadenylation [27], and mRNA degradation [28, 29, 30, 31, 32]. Most MPRAs read out the expression of a reporter gene in one of two ways [1]: by quantifying RNA abundance via the sequencing of RNA barcodes that are linked to known variants (RNA-seq MPRAs; Fig. 1c), or by quantifying protein abundance using fluorescence-activated cell sorting (FACS) then sequencing the sorted variants (sort-seq MPRAs; Fig. 1e).

MAVE data can enable rich quantitative modeling of G-P maps. This key point was recognized in some of the earliest work on MAVEs [17, 18] and has persisted as a major theme in MAVE studies [33, 34, 35, 25, 31, 36]. But in contrast to MAVE experimental techniques, which continue to advance rapidly, there remain key gaps in the methodologies available for quantitatively modeling G-P maps from MAVE data.

Most computational methods for analyzing MAVE data have focused on accurately quantifying the activities of individual assayed sequences [37, 38, 39, 40, 41, 42, 43]. However, MAVE measurements for individual sequences often cannot be interpreted as providing direct quantification of the underlying G-P map that one is interested in. First, MAVE measurements are usually distorted by strong nonlinearities and noise, and distinguishing interesting properties of G-P maps from these confounding factors is not straight-forward. Second, MAVE data is often incomplete. Missing data is common, but a more fundamental issue is that researchers often want to understand G-P maps over vastly larger regions of sequence space than can be exhaustively assayed.

Quantitative modeling can address both the incompleteness and indirectness of MAVE measurements [1]. The goal here is to determine a mathematical function that, given any sequence as input, will return a quantitative value for that sequence's molecular phenotype. Quantitative models thus fill in the gaps in G-P maps and, if appropriate inference methods are used, can further remove confounding effects of nonlinearities and noise. The simplest quantitative modeling

[*] Correspondence: jkinney@cshl.edu
[1] Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, 11375, Cold Spring Harbor, NY
Full list of author information is available at the end of the article

strategy is linear regression (e.g. [18, 44]). However, linear regression yields valid results only when one's measurements are linear readouts of phenotype and exhibit uniform Gaussian noise. Such assumptions are often violated in dramatic fashion by MAVEs, and failure to account for them can give rise to major artifacts, such as spurious epistatic interactions [45].

Multiple MAVE analysis approaches that can separate the effects of nonlinearities and noise from underlying G-P maps have been reported, but a conceptually unified strategy is still needed. Work in the theoretical evolution literature has focused on a phenomenon called global epistasis (GE), in which measurements reflect a nonlinear function of an underlying latent phenotype [46, 47, 48, 49, 50, 51, 52, 53, 45]. In particular, Otwinowski et al. [46] describe a regression approach in which one parametrically models G-P maps while nonparametrically modeling nonlinearities in the MAVE measurement process. Parallel work in the biophysics literature has focused on developing ways to infer G-P maps from high-throughput data in a manner that is agnostic to the quantitative form of both nonlinearities and noise [54, 55, 56, 17]. This approach focuses on the use of mutual information as an objective function. It arose from techniques in sensory neuroscience [57, 58] that were elaborated and adapted for the analysis of microarray data [59, 56], then applied to MPRAs [33, 34, 17, 18]. However, due to difficulties in estimating mutual information, such analyses of MAVE data have relied on Metropolis Monte Carlo, which (in our experience) is too slow to support widespread adoption. A third thread in the literature has arisen from efforts to apply techniques from deep learning to modeling the genotype-phenotype map [60, 28]. Here the emphasis has been on using the highly expressive nature of deep neural networks to directly model experimental output from input sequences. Yet it has remained unclear how such neural networks might separate out the intrinsic features of GP maps from effects of MAVE measurement processes. This is a manifestation of the neural network interpretability problem, one that that is not addressed by established post-hoc attribution methods [61, 62].

Here we describe a unified conceptual and computational framework for the quantitative modeling of MAVE data, one that unites the three strains of thought described above. As illustrated in Fig. 2, we assume that each sequence has a well-defined latent phenotype, of which the MAVE provides a noisy nonlinear readout. To remove potentially confounding effects due to the MAVE measurement process, we model both the G-P map and the measurement process simultaneously. As described in previous theoretical work [54, 55], this turns the latent phenotype into a type of information bottleneck [63, 64], one that separates the task of compressing sequence-encoded information (the job of the G-P map) from the task of mapping this information to a realistic experimental output (the job of the measurement process). Some ambiguity between the quantitative properties of the G-P map and those of the measurement process inevitably remain, but such ambiguities often affect only a small number of parameters in the underlying G-P map [54, 55].

We also introduce MAVE-NN, a software package that can rapidly execute this type of bottleneck-based inference on large MAVE datasets. MAVE-NN currently supports two inference approaches: GE regression and noise agnostic (NA) regression. GE regression is modeled after the approach of [46]. NA regression is closely related to non-parametric regression using information maximization (see [54, 55]) but is much faster due to its compatibility with backpropagation. MAVE-NN is available as a Python API and is built on top of TensorFlow, allowing the use of the powerful and flexible optimization methods. MAVE-NN is rigorously tested and is easily installed from PyPI using the command `pip install mavenn`. Comprehensive documentation is available at http://mavenn.readthedocs.io.

Results

Inference approaches

MAVE-NN supports the analysis of DNA, RNA, and protein sequences. All sequences must be the same length and, for the resulting models to be interpretable, must satisfy a natural notion of alignment. The two inference methods supported by MAVE-NN, GE regression and NA regression, are illustrated in Fig. 3. Each type of regression can be used to infer three different types of model: additive, neighbor, or pairwise. MAVE-NN represents each sequence as a binary vector $\vec{x}$ of one-hot encoded sequence features. The measurement obtained for each sequence is then assumed to depend on a latent phenotype ϕ, which is assumed to depend the sequence via a linear model $\phi = \vec{\theta} \cdot \vec{x}$, where $\vec{\theta}$ denotes model parameters. In both types of regression, MAVE-NN assumes that the measurement of each sequence is a function of a latent phenotype ϕ. This function is inferred from data using the dense subnetworks shown in Fig. 3a and 3c. Examples of these functions are shown in Fig. 3b and 3d. For additional details, see Methods.

In the following sections, we demonstrate the utility of MAVE-NN on previously published DMS, RNA-seq MPRA, and sort-seq MPRA datasets.

Global epistasis model of protein GB1

Protein G is an immunoglobulin-binding protein expressed in streptococcal bacteria [65] and is a model system to study epistasis, protein folding, and the effects of mutations on protein function that has been investigated multiple times through high-throughput mutagenesis[66, 67, 68]. Here we analyze DMS data from [66]. In this work, the authors made all single and double mutations between 55 positions in the domain of protein G (GB1) that binds to immunoglobulin G (IgG). To quantify the affinity of variant GB1 domains (expressed as individual proteins) for IgG, the authors then used mRNA display, an assay in which variant GB1 proteins were covalently linked to their encoding mRNAs and enriched using IgG beads. Deep sequencing was then used to quantify the enrichment of variant mRNAs, corresponding to the ratio of enriched counts to library counts. Our goal here is to quantitatively model log enrichment as a function of sequence.

There is good reason to suspect there might be strong GE nonlinearities in these DMS data. In the simplest case, we can imagine that the Gibbs free energy of a GB1 variant bound to IgG will be an additive function of GB1 sequence, reflecting no energetic epistasis between positions. However, even if the enrichment of variant mRNAs is performed under equilibrium thermodynamic conditions, quantitative enrichment values will reflect the occupancy of each GB1 variant, which is a sigmoidal function of binding energy. We would thus expect to observe a nonlinear relationship between an additive latent phenotype (binding energy), and the experimental readout (log enrichment). Note that for more complicated scenarios, such as enrichment far from equilibrium, the experimental readout is still likely to be a nonlinear function of the underlying latent phenotype. Then again, even in the improbable case where the biological quantity of interest is linearly related to the experimental readout, GE regression will learn that linear relationship.

Motivated by the argument above, as well as the work of Otwinowski *et al.* [46], we fit an additive GE model to the DMS data from [66]. We note that while [46] also fit a GE model to these data, our goal here is different; we aim to highlight the capabilities of MAVE-NN. There are also important differences in our approach: [46] implemented GE regression using a maximum-likelihood inference procedure based on I-splines basis, whereas MAVE-NN formulates this problem using neural networks, allowing the use of TensorFlow for rapid and robust inference [69]. Moreover, the method from [46] is currently implemented only as a Julia script whereas MAVE-NN is a fully document Python API.

The data we used to fit GE models consisted of 535,918 variant GB1 sequences and their corresponding log enrichment values. MAVE-NN completed GE model inference $\sim$ 10 minutes on a standard laptop computer. The results of this inference, including model predictions, additive weights θ_{ic}^{GE}, and the global epistasis nonlinearity $g(\phi)$ are shown in Fig. 4 (panels d and e) and closely match the results of [46]. Moreover, on simulated GB1 data where we know the true underlying nonlinearity and additive parameters, MAVE-NN is able to accurately recover both θ_{ic}^{GE} and g (Fig. S1).

As a baseline, we compare the resulting GE model to an additive model fit using linear regression. We find that GE regression fits held-out test data much better than linear regression does, yielding $R^2 = 0.94$ vs. $R^2 = 0.86$ (Fig. 4a-b). Plotting linear model predictions vs. GE model predictions suggests that there are systematic nonlinear effects not being captured by the linear model (Fig. 4c). Fig. 4f shows linear model weights (θ_{ic}^{lin}) plotted as a sequence logo. Plotting the values of θ_{ic}^{GE} vs. θ_{ic}^{lin} reveals systematic differences, e.g., there appears to be a cluster of parameters which is assigned roughly the same value by the linear model (close to 0 on the x-axis), but to which the GE model assigns a large range of values (Fig. 4g). This analysis thus illustrates the importance of including a global epistasis nonlinearity when modeling sequence-function relationships from DMS data.

Global epistasis model of splice site activity

Splicing is a key step in the expression of human genes. Mutations at $5'$ splice sites ($5'$ ss), which define the boundaries between upstream exons and downstream introns, often cause exon skipping by disrupting spliceoseome recognition, and can result in disease [70, 71]. Here we analyze data from a massively parallel splicing assay (MPSA, a type of RNA-seq MPRA) that was used to quantify the effects of variant $5'$ ss sequences on exon skipping. Specifically, Wong *et al.* [26] measured the effects of all 32,768 possible 9-nucleotide $5'$ ss sequences in three gene contexts: *BRCA2* exon 17, *SMN1* exon 7, and *IKBKAP* exon 20. Their experimental strategy used a three-exon minigene, in which the $5'$ ss of the central exon was varied. Minigene plasmids were transfected into HeLa cells, bulk RNA was extracted, and exon inclusion was assayed using RT-PCR coupled to high-throughput sequencing (Fig. 1c). From the resulting sequence data, the authors calculated a percent-spliced-in (PSI) value based on the amount of exon inclusion RNA relative to total RNA (Fig. 1d). Our goal is to model PSI as a function of the $5'$ ss sequence.

As in the case of GB1, there is good reason to suspect strong GE nonlinearities in these MPSA data. Splicing is a highly complex process involving more than 200 proteins [72], but a simplified argument for why GE nonlinearities might be present is still illuminating. $5'$ ss RNA sequences directly bind the U1 small nuclear ribonucleoprotein (snRNP), a component of the spliceosome, in the early stages of spliceosome assembly. We can imagine that the Gibbs free energy of the U1 snRNP bound to RNA at the $5'$ ss will be an approximately linear function of the $5'$ ss sequence, whereas PSI will reflect the occupancy of the $5'$ ss by the U1 snRNP. Even this highly simplified model requires a strongly nonlinear relationship between a linear latent phenotype (energy) and the experimental readout (PSI).

Previous comparative genomics studies have observed pairwise nucleotide-position dependencies in $5'$ ss sequences [73, 74, 75]. Additionally, Wong *et al.* [26] reported that fitting a linear pairwise model to their MPSA data (using linear regression) better explained the relationship between PSI and $5'$ ss sequences compared to just an additive model. Motivated by these points, we fit both additive and pairwise GE models to these MPSA data. MAVE-NN was able to infer each of these models in ~ 5 minutes. For comparison, we also fit additive and pairwise models using linear regression. All MPSA results, including model predictions and global epistasis nonlinearities, are shown in Fig. 5. Note that we fit our models to $\log_{10}$ PSI values, as these transformed values exhibit approximately uniform noise.

We find that an additive model fit using linear regression produces an R^2 value of only 0.22 on held-out test data, while a linear pairwise model improves that value substantially (Fig. 5a-b). Both linear models, however, show strongly nonlinear patterns of residuals, indicating substantial model misspecification. The additive GE model outperforms both models fit by linear regression (Fig. 5c), and exhibits a highly nonlinear link function $g(\phi)$ (Fig. 5d). However, there are a number of measurements that deviate far from the flat part of the function $g(\phi)$, suggesting that the additive model for the latent phenotype might not be sufficient for explaining these data. This is confirmed by Fig. 5e, which shows that the pairwise GE model substantially outperforms all the other models. Fig. 5f illustrates this improved fit: unlike in panel d, the cluster of points above the flat part of the function $g(\phi)$ now cluster tightly to the pairwise global epistasis nonlinearity. The "neck" of points near the transition region of the nonlinearity is also much thinner. Using simulated pairwise $5'$ ss data, we show that MAVE-NN is able to accurately recover both pairwise latent model and pairwise g accurately (Fig. S2). This analysis highlights the importance of including global epistasis nonlinearities even when pairwise features are included.

Noise agnostic regression on *lac* promoter data

The *lac* promoter of *Escherichia coli* has long served as a model system for studying transcriptional regulation. Kinney *et al.* [17] used this system to demonstrate a massively parallel reporter assay called Sort-Seq (Fig. 1e), which was the first MAVE developed for studies in living cells. The authors created a library of *lac* promoters mutagenized within a 75 bp region that binds two transcription factors, CRP and σ^{70} RNA polymerase (RNAP) [76]. These variant promoters were then used to drive the expression of GFP. Cells containing expression constructs were sorted according to GFP fluorescence into 5-10 bins, and the variant promoters within each bin were sequenced. The resulting data consisted of a list of unique promoter variants along with the number of times each variant was observed in each bin (Fig. 1f). The authors performed a total of six such experiments, using different promoter libraries, host strains, and growth conditions. From these data, they were able to infer precise additive models for the in vivo sequence-dependent binding energies of CRP and RNAP.

Kinney *et al.* [17] fit binding energy models to Sort-Seq data using an inference method called information maximization (IM) regression. Specifically, they searched for additive phenotype parameters that would maximize the mutual information $I[\text{bin};\phi]$ between the phenotype value ϕ (which they interpreted as binding energy) and the bin in which each sequence was found. This approach has a strong theoretical rationale when fitting models to data with uncertain noise characteristics [56, 55, 54]. But in practice, IM regression presents multiple challenges. One difficulty is estimating $I[\text{bin};\phi]$ from finite data. While there are many approaches for estimating mutual information, these methods are relatively slow and IM inference requires doing this after each update of the model parameters. Moreover, Kinney *et al.* [17] and subsequent work [33, 34, 35] have performed IM inference using custom Metropolis Monte Carlo implementations that are slow and which have not been deployed as robust general-use software.

For data like those reported in [17], which consist of sequences and associated counts across multiple bins, MAVE-NN uses an inference strategy that is closely related to but distinct from IM inference: MAVE-NN performs a semiparametric optimization of log likelihood over both the parameters of ϕ and the experimental noise model $\pi(\text{bin}|\phi)$, which represents a probability distribution over bins conditioned on the latent phenotype. We call this approach NA regression.

The connection between IM and NA regression has been elaborated in previous work [55, 54], and is summarized here in the Supplemental Information. Importantly, NA regression is readily formulated using neural networks and thus carried out using stochastic gradient descent within TensorFlow. This enables robust optimization and dramatically reduces computation times.

Here we demonstrate NA regression on the Sort-Seq data of [17] by inferring additive models for the sequence-dependent activity at the RNAP binding site. The results are shown in Fig. 6. Each row in Fig. 6 represents a different Sort-Seq experiment; these five experiments assayed different variant libraries under different experimental conditions (see Supplemental Information for details). In each row, the left-most panel shows sequence logos representing additive models inferred by NA regression. Sequence logos in Fig. 4 and Fig. 6 were made using Logomaker [77]. Center panels illustrate the corresponding noise models $\pi(\text{bin}|\phi)$. Although these noise models differ greatly from experiment to experiment, the parameters $\vec{\theta}$ describing RNAP binding site strength are remarkably consistent with each other and with the known bipartite structure of the RNAP binding motif.

The right-most panels in Fig. 6 plot the NA-inferred parameters $\vec{\theta}$ against those reported by Kinney *et al.* [17] using IM regression. These plots reveal a high level of correspondence, but they still leave open the question of which set of parameters perform better. To address this question, we estimated the mutual information $I[\text{bin};\phi]$ for binding models inferred by both approaches (see Methods). The computed $I[\text{bin};\phi]$ values are displayed within each scatter plot. Although mutual information values for IM and NA regression are comparable, NA regression consistently achieves higher $I[\text{bin};\phi]$, suggesting better performance. Moreover, NA regression, as implemented by MAVE-NN, dramatically reduces the inference time compared to IM regression computed using Metropolis Monte Carlo: NA regression takes a few tens of seconds on a standard laptop computer to infer each of these RNAP models whereas IM regression takes several hours [unpublished data]. Finally, using simulated data, we show that MAVE-NN (using NA regression) recovers ground-truth RNAP additive parameters nearly perfectly (see Fig. S3).

Discussion

In this work we have presented MAVE-NN, a software package for inferring quantitative models of genotype-phenotype (G-P) maps from diverse MAVE datasets. At the core of MAVE-NN is the conceptualization of G-P maps as a type of information bottleneck [64, 63]. Specifically, MAVE-NN assumes that, in a MAVE experiment, the underlying G-P map first compresses an input sequence into a single meaningful scalar – the latent phenotype – and that this quantity is read out only indirectly by a noisy and nonlinear measurement process. By explicitly modeling this measurement process along with the G-P map, MAVE-NN is able to remove potentially confounding effects from the G-P map. We have demonstrated this capability in the context of three diverse MAVE experiments: a deep mutational scanning assay [66], an RNA-seq-based massively parallel splicing assay [26], and a FACS-based massively parallel reporter assay [17]. We have also benchmarked the performance of MAVE-NN on analogous simulated data.

MAVE-NN currently supports two inference methods: GE regression, which is suitable for datasets with continuous target variables and uniform Gaussian noise, and NA regression, which is suitable for datasets with categorical target variables. MAVE-NN also supports three types of G-P models: additive, neighbor, and pairwise. The information bottleneck strategy behind MAVE-NN is very general, however, and we anticipate expanding and generalizing the capabilities of MAVE-NN in the near future.

MAVE-NN is implemented in TensorFlow, which makes the underlying computations fast and robust. It also has an easy-to-use Python API, is thoroughly tested, and can be installed from PyPI by executing "`pip install mavenn`". Comprehensive documentation as well as examples and step-by-step tutorials are available at http://mavenn.readthedocs.io.

Methods

We represent each MAVE dataset as a set of N observations, $\{(\vec{x}_n, y_n)\}_{n=1}^{N}$, where each observation consists of a sequence $\vec{x}_n$ and a measurement y_n of sequence activity. Here, y_n can be either a continuous real-valued number, or a categorical variable representing the bin in which the nth sequence was found. Note that, in this representation, the same sequence $\vec{x}$ can be observed multiple times in each dataset and be assigned different values for y each time due to experimental noise. Datasets with real-valued measurements y are analyzed using GE regression, while datasets with categorical y values are analyzed using NA regression. Both types of regression assume that y_n is a noisy indirect readout of some latent phenotype $\phi(\vec{x}_n;\vec{\theta})$, which is a linear function of its parameters $\vec{\theta}$.

Latent phenotype models

We assume that all sequences have the same length L, and that at each of the L positions in each sequence there is one of C possible characters ($C = 4$ for DNA and RNA; $C = 21$ for protein, representing 20 amino acids and the termination signal). In what follows, each sequence $\vec{x}$ is represented as a binary $C \times L$ matrix having elements

$$x_{cl} = \begin{cases} 1 & \text{if character } c \text{ occurs at position } l \\ 0 & \text{otherwise} \end{cases} . \quad (1)$$

Here, $l = 1, 2, \ldots, L$ indexes positions within the sequence, while c indexes possible nucleotides or amino acids.

Our goal is to derive a function that can, given a sequence $\vec{x}$, predict the value of the latent phenotype that was indirectly measured by the experiment. To do this we assume that this phenotype is given by a function $\phi(\vec{x}; \vec{\theta})$ that depends on the sequence $\vec{x}$ and a set of parameters $\vec{\theta}$. MAVE-NN supports three functional forms for ϕ: an "additive" model, where each position in $\vec{x}$ contributes independently to the latent phenotype,

$$\phi_{\text{additive}}(\vec{x}; \vec{\theta}) = \sum_{l=1}^{L} \sum_{c} \theta_{cl} x_{cl}, \quad (2)$$

a "neighbor" model, which accounts for potential epistatic interactions between neighboring positions,

$$\phi_{\text{neighbor}}(\vec{x}; \vec{\theta}) = \sum_{l=1}^{L-1} \sum_{c,c'} \theta_{cc'l} x_{cl} x_{c'(l+1)}. \quad (3)$$

and a "pairwise" model, which includes interactions between all pairs of positions,

$$\phi_{\text{pairwise}}(\vec{x}; \vec{\theta}) = \sum_{l=1}^{L-1} \sum_{l'=l+1}^{L} \sum_{c,c'} \theta_{cc'll'} x_{cl} x_{c'l'}. \quad (4)$$

All three types of latent phenotype models can be inferred using either GE regression or NA regression.

Global epistasis (GE) regression

GE models assume that each measurement y of a sequence $\vec{x}$ is a nonlinear monotonic function $g(\cdot)$ of a latent phenotype ϕ plus uniform Gaussian noise ϵ:

$$y = g(\phi(\vec{x})) + \epsilon. \quad (5)$$

Given a MAVE dataset $\{(\vec{x}_n, y_n)\}_{n=1}^{N}$, the global epistasis non-linearity $g(\cdot)$ and the linear model parameters $\vec{\theta}$ are inferred by fitting a neural network having the architecture shown in Fig. 3a and using a quadratic loss function,

$$\mathcal{L}[\vec{\theta}, g] = \frac{1}{2N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2 \quad \text{where} \quad \hat{y}_n = g(\phi(\vec{x}_n; \vec{\theta})). \quad (6)$$

Noise agnostic (NA) regression

In NA regression, we assume that the measurement y of a sequence $\vec{x}$ is governed by a noise model $\pi(y|\phi)$, which represents a probability distribution over possible bins y conditioned on a deterministic latent phenotype $\phi(\vec{x}, \vec{\theta})$. The defining feature of NA regression is that the noise model π is not assumed a priori, as it is in standard likelihood-based inference, but rather is inferred from data concurrently with the latent phenotype parameters $\vec{\theta}$. Specifically, given a MAVE dataset $\{(\vec{x}_n, y_n)\}_{n=1}^{N}$, the noise model π and parameters $\vec{\theta}$ are inferred by fitting a neural network, having the architecture shown in Fig. 3c, using the log likelihood objective function,

$$\mathcal{L}[\vec{\theta}, \pi] = \frac{1}{N} \sum_{m=1}^{M} \sum_{y} c_{my} \log[\pi(y|\phi(\vec{x}_m; \vec{\theta}))]. \quad (7)$$

Here, $m = 1, 2, \ldots, M$ indexes unique sequences $\vec{x}_m$, y indexes possible bins, and c_{my} represents the total number of counts observed for the sequence $\vec{x}_m$ in bin y. Note that M and N are related via $N = \sum_{m=1}^{M} \sum_y c_{my}$.

Computation of mutual information

We use the mutual information between bin y and latent phenotype ϕ to evaluate the performance of NA regression. This is given by,

$$I[y; \phi] = \sum_y \int d\phi \; p(y, \phi) \log_2 \frac{p(y|\phi)}{p(y)}. \quad (8)$$

The computation of mutual information requires knowing the probability densities in Eq. 8, where $p(y, \phi)$ and its marginal distributions represent what would be observed in the limit of infinite data. We do not have direct access to these distributions, so instead we make the following approximations. First, we approximate expectation values with respect to $p(y, \phi)$ as an average over our N observations,

$$\begin{aligned} \sum_y \int d\phi \; p(y, \phi) f(y, \phi) &\approx \frac{1}{N} \sum_{n=1}^{N} f(y_n, \phi_n) \\ &= \frac{1}{N} \sum_{m=1}^{M} \sum_y c_{my} f(y, \phi_m), \end{aligned} \quad (9)$$

where $\phi_n = \phi(\vec{x}_n;\vec{\theta})$, $\phi_m = \phi(\vec{x}_m;\vec{\theta})$, and $f(y,\phi)$ is any function of interest. Additionally, we approximate $p(y|\phi)$ by the inferred noise model $\pi(y|\phi)$, and $p(y)$ by $\pi(y) = N^{-1}\sum_{m=1}^{M} c_{my}$, the total fraction of counts in bin y. Putting these together gives

$$I[y;\phi] \approx \frac{1}{N}\sum_{m=1}^{M}\sum_{y} c_{my}\log_2\frac{\pi(y|\phi_m)}{\pi(y)}. \qquad (10)$$

We used Eq. 10 to compute the mutual information reported in Fig. 6.

We note that our method for approximating mutual information differs from that used in [17]. Specifically, Kinney *et al.* [17] assigned a rank order R to each of their model predictions ϕ, then for each value of y they estimated the joint distribution $p(y,R)$ by smoothing with a Gaussian kernel in the R-direction. This smoothed $p(y,R)$ was then used to estimate the mutual information $I[y;R]$ which, due to the reparameterization invariance of mutual information, was used as an estimate of $I[y;\phi]$.

Funding
This work was supported by NIH grant 1R35GM133777 (awarded to JBK) and NIH Grant 1R35GM133613 (awarded to DMM), an Alfred P. Sloan Research Fellowship (awarded to DMM), a grant from the CSHL/Northwell Health partnership, and funding from the Simons Center for Quantitative Biology at Cold Spring Harbor Laboratory.

Availability of data and materials

- Project: mavenn
- Documentation: mavenn.readthedocs.io
- Programming language: Python
- Installation: `pip install mavenn`
- License: MIT
- Restrictions on use by non-academics: None

Competing interests
The authors declare that they have no competing interests.

Author's contributions
JBK, AT, and DMM conceived the project. AT and JBK wrote the software. AT tested the software and released it as a python package on PYPI. AT, DMM, and JBK wrote the manuscript. WTI wrote a preliminary version of the software. AP performed the gauge fixing analysis. All authors contributed to aspects of the analyses.

Acknowledgements
The authors would like to thank Peter Koo for providing valuable comments.

Author details
[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, 11375, Cold Spring Harbor, NY. [2]Department of Physics, California Institute of Technology, 91125, Pasadena, CA.

References

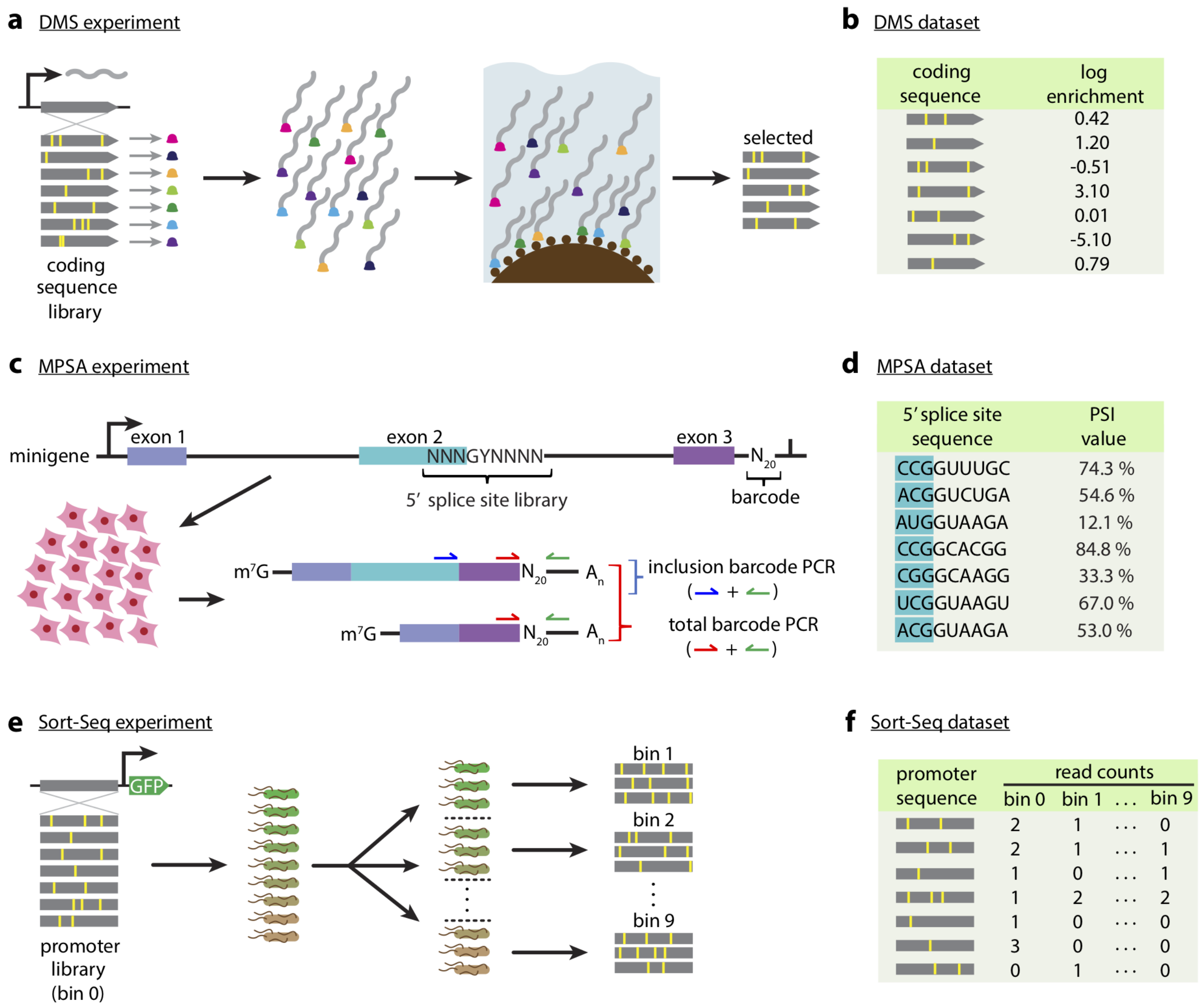
1. Kinney, J.B., McCandlish, D.M.: Massively Parallel Assays and Quantitative Sequence–Function Relationships. Annual Review of Genomics and Human Genetics **20**(1), 99–127 (2019). doi:10.1146/annurev-genom-083118-014845
2. Starita, L.M., Ahituv, N., Dunham, M.J., Kitzman, J.O., Roth, F.P., Seelig, G., Shendure, J., Fowler, D.M.: Variant Interpretation: Functional Assays to the Rescue. American journal of human genetics **101**(3), 315–325 (2017). doi:10.1016/j.ajhg.2017.07.014
3. Esposito, D., Weile, J., Shendure, J., Starita, L.M., Papenfuss, A.T., Roth, F.P., Fowler, D.M., Rubin, A.F.: MaveDB: an open-source platform to distribute and interpret data from multiplexed assays of variant effect. Genome Biology **20**(1), 223 (2019). doi:10.1186/s13059-019-1845-6
4. Fowler, D.M., Fields, S.: Deep mutational scanning: a new style of protein science. Nat Methods **11**(8), 801–807 (2014). doi:10.1038/nmeth.3027
5. Fowler, D.M., Araya, C.L., Fleishman, S.J., Kellogg, E.H., Stephany, J.J., Baker, D., Fields, S.: High-resolution mapping of protein sequence-function relationships. Nature Methods **7**(9), 741–746 (2010)
6. Hietpas, R.T., Jensen, J.D., Bolon, D.N.A.: Experimental illumination of a fitness landscape. Proc Natl Acad Sci USA **108**(19), 7896–7901 (2011)
7. McLaughlin, R.N., Poelwijk, F.J., Raman, A., Gosal, W.S., Ranganathan, R.: The spatial architecture of protein function and adaptation. Nature **491**(7422), 138–142 (2012)
8. Pitt, J.N., Ferré-D?Amaré, A.R.: Rapid construction of empirical rna fitness landscapes. Science **330**(6002), 376–379 (2010)
9. Puchta, O., Cseke, B., Czaja, H., Tollervey, D., Sanguinetti, G., Kudla, G.: Network of epistatic interactions within a yeast snorna. Science **352**(6287), 840–844 (2016)
10. Li, C., Qian, W., Maclean, C.J., Zhang, J.: The fitness landscape of a tRNA gene. Science **352**(6287), 837–840 (2016). doi:10.1126/science.aae0568
11. Domingo, J., Diss, G., Lehner, B.: Pairwise and higher-order genetic interactions during the evolution of a tRNA. Nature **558**(7708), 117–121 (2018). doi:10.1038/s41586-018-0170-7
12. Inoue, F., Ahituv, N.: Decoding enhancers using massively parallel reporter assays. Genomics **106**(3), 159–164 (2015)
13. Levo, M., Segal, E.: In pursuit of design principles of regulatory sequences. Nature reviews Genetics **15**(7), 453–468 (2014). doi:10.1038/nrg3684
14. Peterman, N., Levine, E.: Sort-seq under the hood: implications of design choices on large-scale characterization of sequence-function relations. BMC Genomics **17**(1), 206 (2016). doi:10.1186/s12864-016-2533-5
15. White, M.A.: Understanding how cis-regulatory function is encoded in DNA sequence using massively parallel reporter assays and designed sequences. Genomics **106**(3), 165–170 (2015). doi:10.1016/j.ygeno.2015.06.003
16. Patwardhan, R.P., Lee, C., Litvin, O., Young, D.L., Pe'er, D., Shendure, J.: High-resolution analysis of DNA regulatory elements by synthetic saturation mutagenesis. Nat Biotechnol **27**(12), 1173–1175 (2009)
17. Kinney, J.B., Murugan, A., Callan, C.G., Cox, E.C.: Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence. Proceedings of the National Academy of Sciences **107**(20), 9158–9163 (2010). doi:10.1073/pnas.1004290107
18. Melnikov, A., Murugan, A., Zhang, X., Tesileanu, T., Wang, L., Rogov, P., Feizi, S., Gnirke, A., Callan, C.G., Kinney, J.B., Kellis, M., Lander, E.S., Mikkelsen, T.S.: Systematic dissection and optimization of inducible enhancers in human cells using a massively parallel reporter assay. Nat Biotechnol **30**(3), 271–277 (2012)
19. Patwardhan, R.P., Hiatt, J.B., Witten, D.M., Kim, M.J., Smith, R.P., May, D., Lee, C., Andrie, J.M., Lee, S.-I., Cooper, G.M., Ahituv, N., Pennacchio, L.A., Shendure, J.: Massively parallel functional dissection of mammalian enhancers in vivo. Nat Biotechnol **30**(3), 265–270 (2012)
20. Kwasnieski, J.C., Mogno, I., Myers, C.A., Corbo, J.C., Cohen, B.A.: Complex effects of nucleotide variants in a mammalian cis-regulatory element. Proc Natl Acad Sci USA **109**(47), 19498–19503 (2012)
21. Sharon, E., Kalma, Y., Sharp, A., Raveh-Sadka, T., Levo, M., Zeevi, D., Keren, L., Yakhini, Z., Weinberger, A., Segal, E.: Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters. Nat Biotechnol **30**(6), 521–530 (2012)
22. Cheung, R., Insigne, K.D., Yao, D., Burghard, C.P., Wang, J., Hsiao, Y.-H.E., Jones, E.M., Goodman, D.B., Xiao, X., Kosuri, S.: A

Multiplexed Assay for Exon Recognition Reveals that an Unappreciated Fraction of Rare Genetic Variants Cause Large-Effect Splicing Disruptions. Mol Cell **73**(1), 183–1948 (2019). doi:10.1016/j.molcel.2018.10.037

23. Ke, S., Shang, S., Kalachikov, S.M., Morozova, I., Yu, L., Russo, J.J., Ju, J., Chasin, L.A.: Quantitative evaluation of all hexamers as exonic splicing elements. Genome Res **21**(8), 1360–1374 (2011). doi:10.1101/gr.119628.110
24. Ke, S., Anquetil, V., Zamalloa, J.R., Maity, A., Yang, A., Arias, M.A., Kalachikov, S., Russo, J.J., Ju, J., Chasin, L.A.: Saturation mutagenesis reveals manifold determinants of exon definition. Genome Res **28**(1), 11–24 (2018). doi:10.1101/gr.219683.116
25. Rosenberg, A.B., Patwardhan, R.P., Shendure, J., Seelig, G.: Learning the Sequence Determinants of Alternative Splicing from Millions of Random Sequences. Cell **163**(3), 698–711 (2015)
26. Wong, M.S., Kinney, J.B., Krainer, A.R.: Quantitative activity profile and context dependence of all human 5' splice sites. Molecular Cell **71**(6), 1012–10263 (2018). doi:10.1016/j.molcel.2018.07.033
27. Bogard, N., Linder, J., Rosenberg, A.B., Seelig, G.: A Deep Neural Network for Predicting and Engineering Alternative Polyadenylation. Cell **178**(1), 91–10623 (2019). doi:10.1016/j.cell.2019.04.046
28. Cuperus, J.T., Groves, B., Kuchina, A., Rosenberg, A.B., Jojic, N., Fields, S., Seelig, G.: Deep learning of the regulatory grammar of yeast 5' untranslated regions from 500,000 random sequences. Genome Res **27**(12), 2015–2024 (2017). doi:10.1101/gr.224964.117
29. Dvir, S., Velten, L., Sharon, E., Zeevi, D., Carey, L.B., Weinberger, A., Segal, E.: Deciphering the rules by which 5'-UTR sequences affect protein expression in yeast. Proc Natl Acad Sci USA **110**(30), 2792–801 (2013). doi:10.1073/pnas.1222534110
30. Oikonomou, P., Goodarzi, H., Tavazoie, S.: Systematic identification of regulatory elements in conserved 3' UTRs of human transcripts. Cell Rep **7**(1), 281–292 (2014)
31. Sample, P.J., Wang, B., Reid, D.W., Presnyak, V., McFadyen, I.J., Morris, D.R., Seelig, G.: Human 5' UTR design and variant effect prediction from a massively parallel translation assay. Nature Biotechnology **37**(7), 803–809 (2019). doi:10.1038/s41587-019-0164-5
32. Shalem, O., Sharon, E., Lubliner, S., Regev, I., Lotan-Pompan, M., Yakhini, Z., Segal, E.: Systematic dissection of the sequence determinants of gene 3' end mediated expression control. PLoS genetics **11**(4), 1005147 (2015). doi:10.1371/journal.pgen.1005147
33. Barnes, S.L., Belliveau, N.M., Ireland, W.T., Kinney, J.B., Phillips, R.: Mapping DNA sequence to transcription factor binding energy in vivo. PLOS Computational Biology **15**(2), 1006226 (2019). doi:10.1371/journal.pcbi.1006226
34. Belliveau, N.M., Barnes, S.L., Ireland, W.T., Jones, D.L., Sweredoski, M.J., Moradian, A., Hess, S., Kinney, J.B., Phillips, R.: Systematic approach for dissecting the molecular mechanisms of transcriptional regulation in bacteria. Proceedings of the National Academy of Sciences **115**(21), 201722055 (2018). doi:10.1073/pnas.1722055115
35. Boer, C.G.d., Vaishnav, E.D., Sadeh, R., Abeyta, E.L., Friedman, N., Regev, A.: Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. Nature Biotechnology, 1–10 (2019). doi:10.1038/s41587-019-0315-8
36. Kemble, H., Nghe, P., Tenaillon, O.: Recent insights into the genotype–phenotype relationship from massively parallel genetic assays. Evolutionary applications **12**(9), 1721–1742 (2019)
37. Fowler, D.M., Araya, C.L., Gerard, W., Fields, S.: Enrich: software for analysis of protein function by enrichment and depletion of variants. Bioinformatics **27**(24), 3430–3431 (2011)
38. Alam, K.K., Chang, J.L., Burke, D.H.: FASTAptamer: A Bioinformatic Toolkit for High-throughput Sequence Analysis of Combinatorial Selections. Mol Ther Nucleic Acids **4**(3), 230 (2015)
39. Bloom, J.D.: Software for the analysis and visualization of deep mutational scanning data. BMC Bioinformatics **16**, 168 (2015)
40. Rubin, A.F., Gelman, H., Lucas, N., Bajjalieh, S.M., Papenfuss, A.T., Speed, T.P., Fowler, D.M.: A statistical framework for analyzing deep mutational scanning data. Genome Biology **18**(1), 150 (2017). doi:10.1186/s13059-017-1272-5
41. Ashuach, T., Fischer, D.S., Kreimer, A., Ahituv, N., Theis, F.J., Yosef, N.: Mpranalyze: statistical framework for massively parallel reporter assays. Genome Biology **20**(1), 183 (2019). doi:10.1186/s13059-019-1787-z
42. Niroula, A., Ajore, R., Nilsson, B.: MPRAscore: robust and non-parametric analysis of massively parallel reporter assays. Bioinformatics **35**(24), 5351–5353 (2019). doi:10.1093/bioinformatics/btz591. https://academic.oup.com/bioinformatics/article-pdf/35/24/5351/31797907/btz591.pdf
43. Faure, A.J., Schmiedel, J.M., Baeza-Centurion, P., Lehner, B.: Dimsum: an error model and pipeline for analyzing deep mutational scanning data and diagnosing common experimental pathologies. bioRxiv (2020). doi:10.1101/2020.06.25.171421. https://www.biorxiv.org/content/early/2020/06/26/2020.06.25.171421.full.pdf
44. Myint, L., Avramopoulos, D.G., Goff, L.A., Hansen, K.D.: Linear models enable powerful differential activity analysis in massively parallel reporter assays. BMC Genomics **20**(1), 209 (2019). doi:10.1186/s12864-019-5556-x
45. Baeza-Centurion, P., Miñana, B., Schmiedel, J.M., Valcárcel, J., Lehner, B.: Combinatorial genetics reveals a scaling law for the effects of mutations on splicing. Cell **176**(3), 549–563 (2019)
46. Otwinowski, J., McCandlish, D.M., Plotkin, J.B.: Inferring the shape of global epistasis. Proceedings of the National Academy of Sciences **115**(32), 7550–7558 (2018). doi:10.1073/pnas.1804015115. https://www.pnas.org/content/115/32/E7550.full.pdf
47. Otwinowski, J., Nemenman, I.: Genotype to phenotype mapping and the fitness landscape of the e. coli lac promoter. PloS one **8**(5), 61570 (2013)
48. Sarkisyan, K.S., Bolotin, D.A., Meer, M.V., Usmanova, D.R., Mishin, A.S., Sharonov, G.V., Ivankov, D.N., Bozhanova, N.G., Baranov, M.S., Soylemez, O., Bogatyreva, N.S., Vlasov, P.K., Egorov, E.S., Logacheva, M.D., Kondrashov, A.S., Chudakov, D.M., Putintseva, E.V., Mamedov, I.Z., Tawfik, D.S., Lukyanov, K.A., Kondrashov, F.A.: Local fitness landscape of the green fluorescent protein. Nature **533**(7603), 397–401 (2016)
49. Pokusaeva, V.O., Usmanova, D.R., Putintseva, E.V., Espinar, L., Sarkisyan, K.S., Mishin, A.S., Bogatyreva, N.S., Ivankov, D.N., Akopyan, A.V., Avvakumov, S.Y., Povolotskaya, I.S., Filion, G.J., Carey, L.B., Kondrashov, F.A.: An experimental assay of the interactions of amino acids from orthologous sequences shaping a complex fitness landscape. PLoS genetics **15**(4), 1008079–1008079 (2019). doi:10.1371/journal.pgen.1008079
50. Sailer, Z.R., Harms, M.J.: Detecting high-order epistasis in nonlinear genotype-phenotype maps. Genetics **205**(3), 1079–1088 (2017). doi:10.1534/genetics.116.195214. https://www.genetics.org/content/205/3/1079.full.pdf
51. Sailer, Z.R., Harms, M.J.: Uninterpretable interactions: epistasis as uncertainty. BioRxiv, 378489 (2018)
52. Fernandez-de-Cossio-Diaz, J., Uguzzoni, G., Pagnani, A.: Unsupervised inference of protein fitness landscape from deep mutational scan. bioRxiv (2020). doi:10.1101/2020.03.18.996595. https://www.biorxiv.org/content/early/2020/03/20/2020.03.18.996595.full.pdf
53. Domingo, J., Baeza-Centurion, P., Lehner, B.: The causes and consequences of genetic interactions (epistasis). Annual Review of Genomics and Human Genetics **20**(1), 433–460 (2019). doi:10.1146/annurev-genom-083118-014857. PMID: 31082279. https://doi.org/10.1146/annurev-genom-083118-014857
54. Atwal, G.S., Kinney, J.B.: Learning quantitative sequence–function relationships from massively parallel experiments. Journal of Statistical Physics **162**(5), 1203–1243 (2016). doi:10.1007/s10955-015-1398-3
55. Kinney, J.B., Atwal, G.S.: Parametric inference in the large data limit using maximally informative models. Neural Computation **26**(4), 637–653 (2014). doi:10.1162/NECO_a_00568. PMID: 24479782
56. Kinney, J.B., Tkačik, G., Callan, C.G.: Precise physical models of protein–DNA interaction from high-throughput data. Proceedings of the National Academy of Sciences **104**(2), 501–506 (2007). doi:10.1073/pnas.0609908104
57. Sharpee, T., Rust, N., Bialek, W.: Analyzing neural responses to natural signals: maximally informative dimensions. Neural Comput

16(2), 223–250 (2004)

58. Sharpee, T., Sugihara, H., Kurgansky, A., Rebrik, S., Stryker, M., Miller, K.: Adaptive filtering enhances information transmission in visual cortex. Nature **439**(7079), 936–942 (2006). doi:10.1038/nature04519
59. Elemento, O., Slonim, N., Tavazoie, S.: A universal framework for regulatory element discovery across all genomes and data types. Mol Cell **28**(2), 337–350 (2007). doi:10.1016/j.molcel.2007.09.027
60. Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J.: Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. Nature biotechnology **33**(8), 831–838 (2015)
61. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. ICML'17, pp. 3145–3153 (2017)
62. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
63. Tishby, N., Zaslavsky, N.: Deep Learning and the Information Bottleneck Principle. arXiv (2015). 1503.02406
64. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. arXiv **arXiv:physics/0004057 [physics.data-an]**(7), 1601–1612 (2000). doi:10.1109/tip.2009.2017823
65. Sjöbring, U., Bjorck, L., Kastern, W.: Streptococcal protein g. gene structure and protein binding properties. Journal of Biological Chemistry **266**(1), 399–405 (1991). http://www.jbc.org/content/266/1/399.full.pdf+html
66. Olson, C.A., Wu, N., Sun, R.: A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. Current Biology **24**(22), 2643–2651 (2014). doi:10.1016/j.cub.2014.09.072
67. Wu, N.C., Dai, L., Olson, C.A., Lloyd-Smith, J.O., Sun, R.: Adaptation in protein fitness landscapes is facilitated by indirect paths. Elife **5**, 16965 (2016)
68. Nisthal, A., Wang, C.Y., Ary, M.L., Mayo, S.L.: Protein stability engineering insights revealed by domain-wide comprehensive mutagenesis. Proceedings of the National Academy of Sciences **116**(33), 16367–16377 (2019)
69. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., *et al.*: Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283 (2016)
70. Krawczak, M., Reiss, J., Cooper, D.N.: The mutational spectrum of single base-pair substitutions in mRNA splice junctions of human genes: causes and consequences. Hum Genet **90**(1-2), 41–54 (1992). doi:10.1007/bf00210743
71. Srebrow, A., Kornblihtt, A.R.: The connection between splicing and cancer. J Cell Sci **119**(Pt 13), 2635–2641 (2006). doi:10.1242/jcs.03053
72. Will, C.L., Lührmann, R.: Spliceosome structure and function. Cold Spring Harb Perspect Biol **3**(7) (2011). doi:10.1101/cshperspect.a003707
73. Burge, C., Karlin, S.: Prediction of complete gene structures in human genomic dna. J Mol Biol **268**(1), 78–94 (1997). doi:10.1006/jmbi.1997.0951
74. Carmel, I., Tal, S., Vig, I., Ast, G.: Comparative analysis detects dependencies among the 5' splice-site positions. RNA **10**(5), 828–840 (2004). doi:10.1261/rna.5196404
75. Roca, X., Olson, A.J., Rao, A.R., Enerly, E., Kristensen, V.N., Børresen-Dale, A.-L., Andresen, B.S., Krainer, A.R., Sachidanandam, R.: Features of 5'-splice-site efficiency derived from disease-causing mutations and comparative genomics. Genome Res **18**(1), 77–87 (2008). doi:10.1101/gr.6859308
76. Ptashne, M., Gann, A.: Genes & Signals vol. 402. Cold Spring Harbor Laboratory Press Cold Spring Harbor, NY, ??? (2002)
77. Tareen, A., Kinney, J.B.: Logomaker: beautiful sequence logos in Python. Bioinformatics **36**(7), 2272–2274 (2019). doi:10.1093/bioinformatics/btz921. https://academic.oup.com/bioinformatics/article-pdf/36/7/2272/33027659/btz921.pdf



coding sequence	log enrichment
	0.42
	1.20
	-0.51
	3.10
	0.01
	-5.10
	0.79

5′ splice site sequence	PSI value
CCGGUUUGC	74.3 %
ACGGUCUGA	54.6 %
AUGGUAAGA	12.1 %
CCGGCACGG	84.8 %
CGGGCAAGG	33.3 %
UCGGUAAGU	67.0 %
ACGGUAAGA	53.0 %

promoter sequence	read counts			
	bin 0	bin 1	. . .	bin 9
	2	1	. . .	0
	2	1	. . .	1
	1	0	. . .	1
	1	2	. . .	2
	1	0	. . .	0
	3	0	. . .	0
	0	1	. . .	0

Figure 1 Three different multiplex assays of variant effect (MAVEs) and their resulting datasets. (a) The deep mutational scanning (DMS) assay of [66]. Randomly mutagenized gene sequences (gray) produced variant proteins (colored bells). To quantify the affinity of variant GB1 sequences for IgG, mRNA display was used in which variant GB1 proteins were covalently linked to their encoding mRNAs (silver wavy lines) and enriched using IgG beads. Deep sequencing was then used to quantify the enrichment of variant mRNAs, corresponding to the ratio of enriched counts to library counts. From these read counts, a log enrichment value was assigned to each sequence. (b) The dataset produced by this DMS assay consists of variant protein coding sequences and their corresponding log enrichment values. (c) The massively parallel splicing assay (MPSA) assay of [26]. A 3-exon minigene library was generated in which, for each minigene, the $5'$ ss sequence of exon 2 was replaced by a randomized 9-nt sequence, and a unique 20-nt barcode was inserted into the $3'$ UTR. This library was transfected into HeLa cells, followed by RNA extraction and reverse transcription. Barcodes from two different classes of mRNA were then amplified and sequenced: (i) mRNA that included exon 2 and (ii) total mRNA. PSI was calculated using the ratio of inclusion to total barcode counts, normalized to the value obtained for the consensus $5'$ ss sequence (CAGGUAAGU). Colored half arrows represent PCR primers. (d) The data produced by this MPSA consists of randomized splice site sequences and their corresponding PSI values. (e) The Sort-Seq assay of [17]. A plasmid library was generated in which mutagenized versions of a bacterial promoter drive the expression of a fluorescent protein. Cells carrying these plasmids were then sorted according to measured fluorescence using fluorescence-activated cell sorting (FACS). The variant promoters in each bin of sorted cells were then sequenced. (f) The Sort-Seq dataset consists of variant promoter sequences and their read counts in the input library (bin 0) and the nine output bins (1-9).

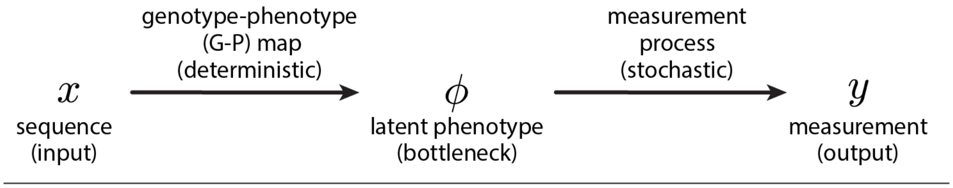


Figure 2 Quantitative modeling strategy. Each input sequence x is mapped to a latent phenotype ϕ via a deterministic genotype-phenotype (G-P) map. A stochastic measurement process then maps ϕ to a observed measurement y. Model inference consists of identifying a G-P map, as well as a measurement process, that together explain the (x, y) pairs in a MAVE dataset. It is the role of ϕ as an information bottleneck that allows some (but not all) properties of the G-P map to be distinguished from properties of the measurement process [54, 55].

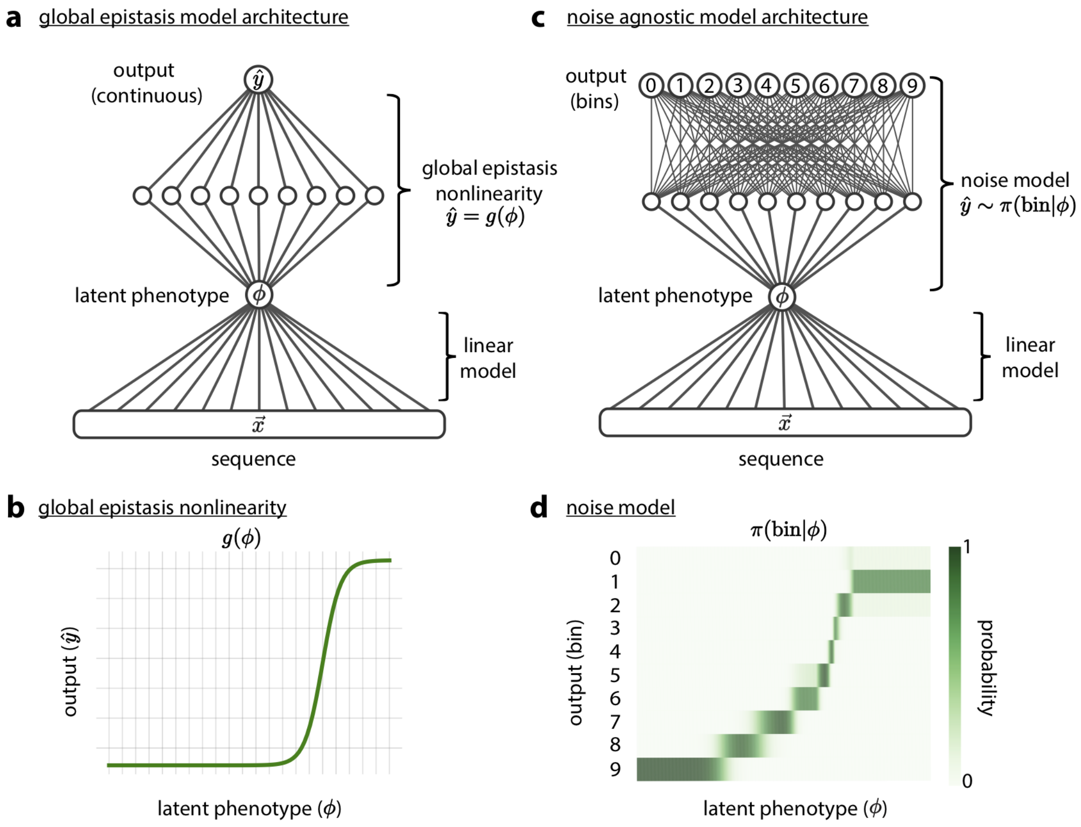


Figure 3 MAVE-NN model architectures. (a) Global epistasis model architecture[48]. A one-layer neural network is used to model the linear dependence of molecular phenotype ϕ on sequence $\vec{x}$. A global epistasis nonlinearity g, which maps ϕ to output $\hat{y}$, is modeled as a dense subnetwork with a nonlinearly activated hidden layer and a linearly activated output node $\hat{y}$. (b) The output $\hat{y}$ of the global epistasis model is a nonlinear function of the latent phenotype ϕ. (c) Noise agnostic model architecture. A one-layer neural network is used to model the linear dependence of ϕ on sequence $\vec{x}$. This value ϕ is then fed to the noise model, a neural network network with a nonlinearly activated hidden layer and a softmax output layer. (d) The noise model, $\pi(\text{bin}|\phi)$, represents a probability distribution over bins conditioned on ϕ.

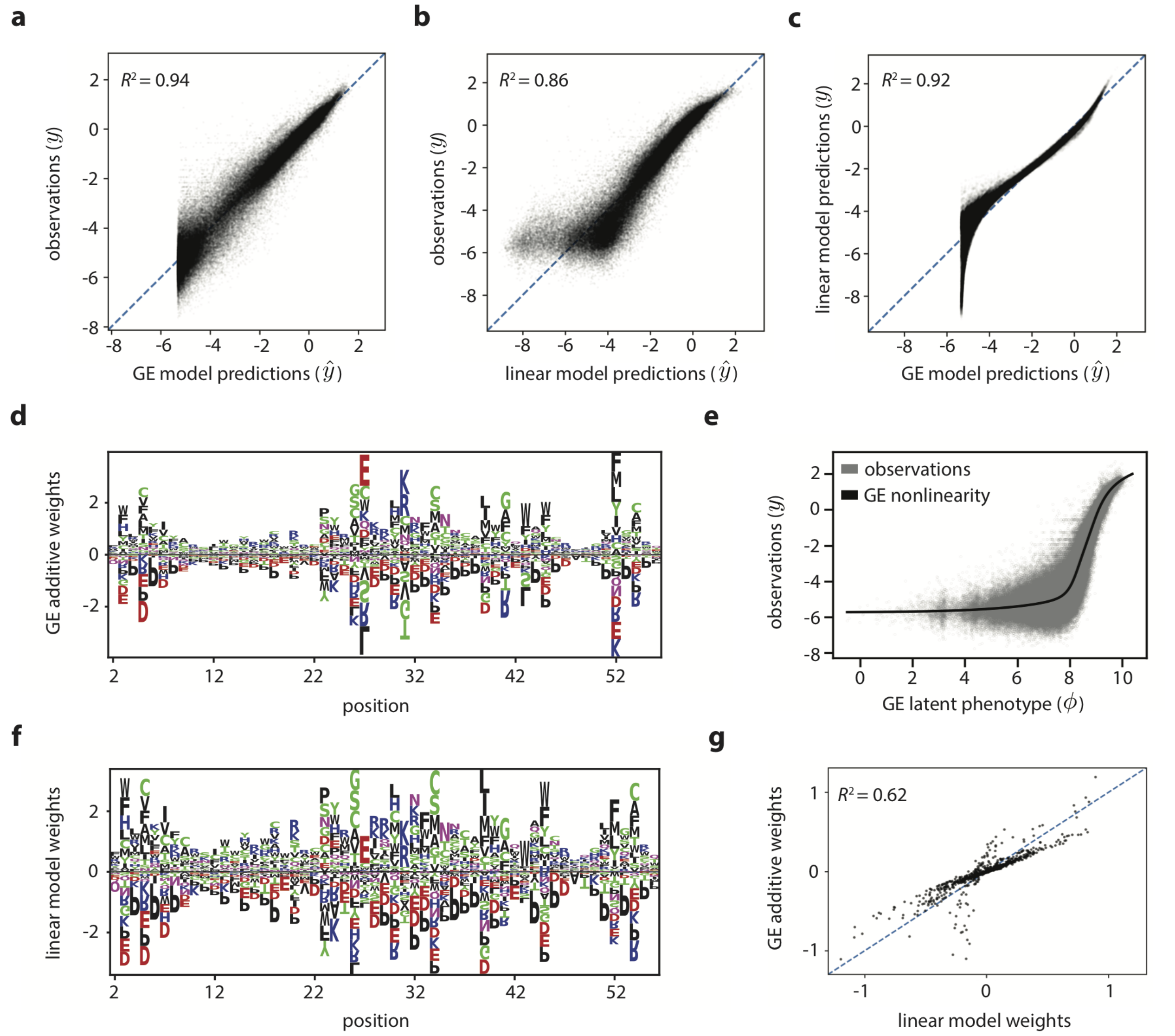


Figure 4 GE regression vs. linear regression on DMS data from [66]. (a) GE regression achieves $R^2 = 0.94$ on held-out test data. (b) Linear regression on the same data yields only $R^2 = 0.86$. (c) Plotting linear model predictions vs. GE model predictions reveals systematic nonlinear deviations. (d) Sequence logo showing additive model weights (θ_{ic}^{GE}) extracted from the first layer of the GE model. (e) Observations (gray) versus GE latent phenotype values. The GE nonlinearity $g(\phi)$ is shown in black. (f) Sequence logo showing the weights (θ_{ic}^{lin}) inferred using linear regression. (g) Comparison of linear model weights and GE additive weights yields $R^2 = 0.62$. The data used to fit these models consisted of 535,918 variant GB1 sequences and their corresponding log enrichment values, which were split $60\% : 20\% : 20\%$ into training, validation, and test sets. We chose a global epistasis architecture with 1 sigmoidally activated hidden layer comprising 200 nodes. The first layer consists of 1100 parameters which constitute the additive model weights. The nonlinear subnetwork consists of 600 parameters, with the total number of parameters in the model equaling 1702 (including biases). We used the Adam optimizer with a learning rate of 5×10^{-4} and used mean squared error as the loss function. All predictions ($\hat{y}$), observations (y), and R^2 values are reported using held-out test data. Sequence logos in panels (d, f) are mean centered and variance normalized.

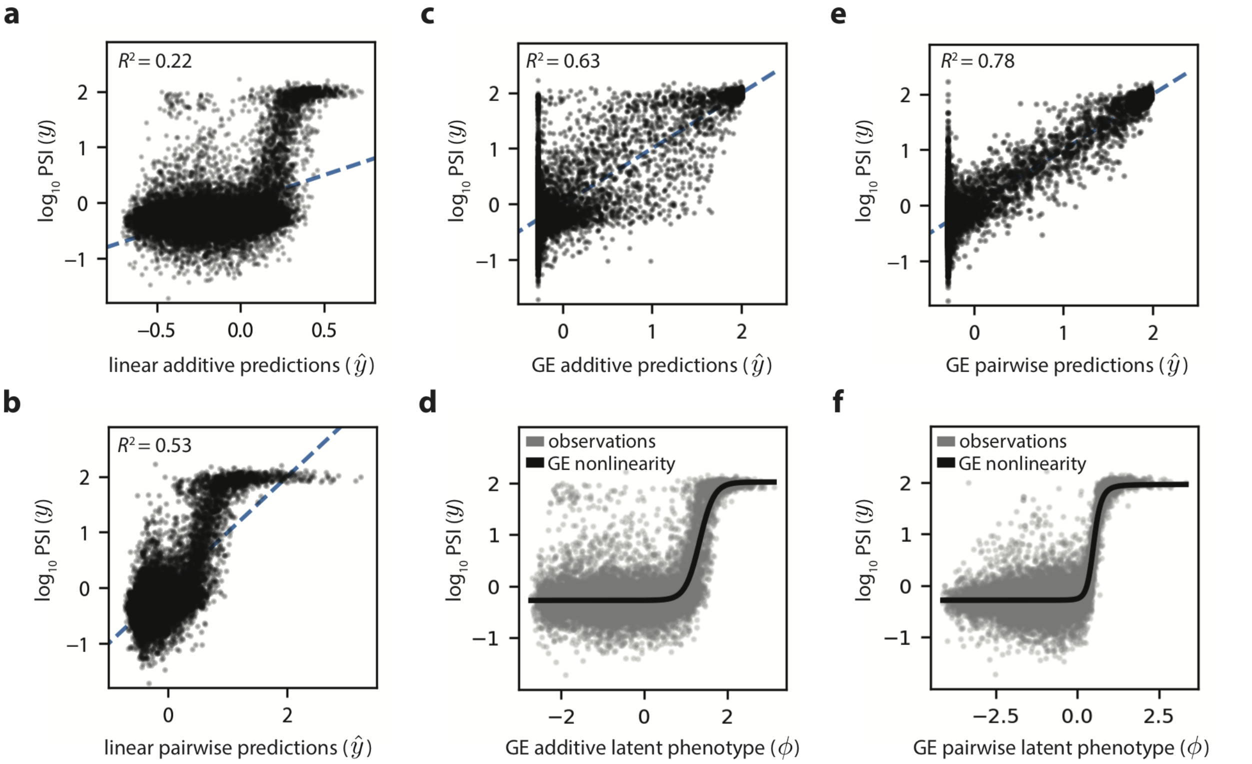


Figure 5 Four models fit to the *BRCA2* MPSA data from [26]. (a) Linear regression using additive sequence features yields only $R^2 = 0.22$ on held out test data. (b) Using pairwise sequence features as input to linear regression substantially improves R^2 on the same test data. (c) GE regression with additive input features achieves $R^2 = 0.63$. (d) Observations (gray) versus GE additive latent phenotype values. The GE additive nonlinearity $g(\phi)$ is shown in black. (e) GE regression with pairwise input features achieves $R^2 = 0.78$, substantially improving performance compared to the additive GE model and pairwise linear model. (f) Observations (gray) versus pairwise GE latent phenotype values. The data used to fit these models are illustrated in Fig. 1d. They consist of 32,768 randomized $5'$ ss sequences and their corresponding measurements ($y = \log_{10}$ PSI), which were split $60\% : 20\% : 20\%$ into training, validation, and test sets. For both additive and pairwise GE models, we chose a global epistasis architecture with 1 sigmoidally activated hidden layer comprising 50 nodes. Weights connecting the input layer to the latent phenotype layer, which constitute the linear model, consist of 36 parameters, with the total number of parameters of the entire model equaling 188 (including biases). We used the Adam optimizer with a learning rate of 10^{-3} and used mean squared error as the loss function. The points plotted in all panels are from the training sets. The R^2 values shown are computed using test sets.

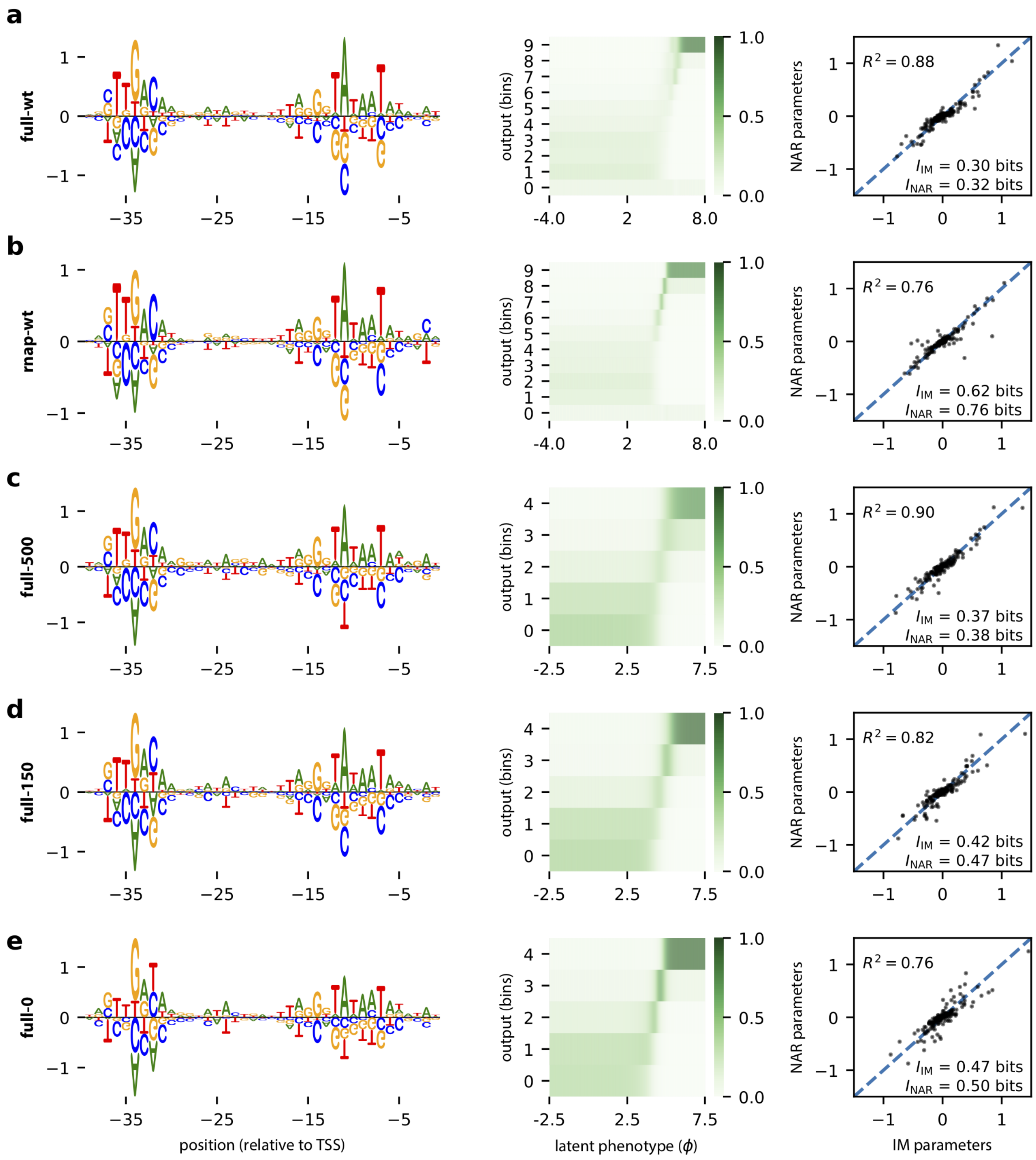


Figure 6 Noise agnostic regression for five different Sort-Seq experiments probing the in vivo binding of *E. coli* σ^{70} RNA polymerase (RNAP) [17]. Each panel presents results from one experiment: (a) full-wt, (b) rnap-wt, (c) full-500, (d) full-150, and (e) full-0; see S.I. for descriptions of experimental conditions. Shown are the NAR parameters θ_{ic}^{NAR} (sequence logo), the inferred noise model $P(\text{bin}|\phi)$ (heat map), and a scatter plot of NAR parameters vs. the IM parameters (θ_{ic}^{IM}) reported in [17]. The data used to fit these models are illustrated in Fig. 1f; bin 0 represents the promoter library and higher bin numbers correspond to higher activity values. For all experiments we used a noise model with a sigmoidally activated hidden layer and a softmax-activated output layer, the Adam optimizer with a learning rate of 10^{-3}, log likelihood loss, and an $60\% : 20\% : 20\%$ split into training, validation, and test sets. Mutual information in bits ($I[\text{bin};\phi]$) is shown for models inferred by IM and by NAR. Parameters from both NAR and IM inference are mean centered and variance normalized.