bioRχiv, 2025, 1–8
doi:
Original Article

GaugeFixer: overcoming parameter non-identifiability in models of sequence-function relationships

Carlos Martí-Gómez,[1] David M. McCandlish[1] and Justin B. Kinney[1,*]

[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, 1 Bungtown Rd., Cold Spring Harbor, 11724, New York, United States

*Corresponding author. jkinney@cshl.edu

Abstract

Mathematical models that describe sequence-function relationships are widely used in computational biology. A key challenge when interpreting these models is that their parameters are not uniquely determined: many different parameter choices can encode the same sequence-function landscape. These ambiguities, known as "gauge freedoms," must be resolved before parameter values can be meaningfully interpreted. Resolving gauge freedoms requires imposing mathematical constraints on parameters that remove these degrees of freedom, a procedure called "fixing the gauge." We recently developed mathematical methods for fixing the gauge of a large class of commonly used models, but the direct computational implementation of these methods is often impractical due to the need for projection matrices whose memory requirements scale quadratically with the number of parameters. Here we introduce GaugeFixer, a Python package that exploits the specific mathematical structure of gauge-fixing projections to achieve linear scaling, thus enabling application to models with millions of parameters. As one application, we analyzed the local structure of peaks in an empirical fitness landscape for translation initiation. GaugeFixer reveals striking similarities, but also fine-scaled variation, in ribosome binding preferences at different positions relative to the start codon, thereby aiding the interpretation of an otherwise unwieldy fitness landscape. GaugeFixer thus fills an unmet need in the computational tools available for biologically interpreting sequence-function relationships.

Availability and implementation: GaugeFixer is compatible with Python $\geq$ 3.10 and can be installed using the pip package. Documentation is provided at http://gaugefixer.readthedocs.io. Source code is available at http://github.com/jbkinney/gaugefixer, as are the scripts used to carry out the analyses presented here.

Introduction

Computational biology routinely involves the use of models that describe the quantitative relationship between biological sequences (DNA, RNA, or protein) and measurable biological activities [Kinney and McCandlish, 2019]. For example, quantitative models of sequence-function relationships have been used to predict the locations of transcription factor binding sites in promoters and enhancers [Stormo, 2013], the locations of splice sites in pre-mRNA transcripts [Yeo and Burge, 2004], structural contacts between residues in folded proteins [Marks et al., 2012], and the effects of human genetic variation on protein function [Hopf et al., 2017]. One application of growing interest is the modeling of data from high-throughput mutagenesis experiments [Kinney et al., 2010, Mogno et al., 2013, Belliveau et al., 2018, Otwinowski et al., 2018, Faure et al., 2022, Zhou et al., 2022, Tareen et al., 2022, Faure and Lehner, 2024, Martí-Gómez et al., 2025, Zhou et al., 2025].

A common way to mathematically represent sequence-function relationships is to use generalized one-hot models [Posfai et al., 2025b,a]. In these models, each sequence is represented as a set of binary features, each feature indicating the presence or absence of a specific subsequence at a specific set of positions. For example, additive features indicate individual characters (nucleotides or amino acids) at individual positions, pairwise features indicate pairs of characters at pairs of positions, and so on. Each parameter of a generalized one-hot model quantifies the effect that a corresponding sequence feature has. Although generalized one-hot models are linear functions of their parameters, they can represent arbitrarily complex sequence-function landscapes [Posfai et al., 2025b].

A fundamental challenge often arises when one attempts to interpret the parameters of generalized one-hot models: except in unusual cases, these parameters are not uniquely determined by the sequence-function landscape the model describes [Posfai et al., 2025a]. As a consequence, the values of model parameters cannot be uniquely determined by data, even in principle. This non-identifiability occurs because the binary feature vectors that represent sequences do not span the full space in which they live. Consequently, a nontrivial set of parameter vectors will yield identical model predictions. These parameter vectors differ by what are called "gauge freedoms" – directions in parameter space that are orthogonal to all binary feature vectors.

The ambiguity that gauge freedoms produce must be resolved before the values of model parameters can be interpreted, or before the parameters of two different models can be compared. This resolution is achieved by imposing additional mathematical constraints that, together with the predicted landscape, select a unique set of parameter values. This procedure is called "fixing the gauge". Different choices of constraints yield different gauges, each providing a distinct mathematical representation of the same underlying sequence-function relationship reflecting a distinct interpretation of the parameters.

The importance of gauge freedoms was first recognized in theoretical physics and is very well understood in that context [Jackson and Okun, 2001]. Until recently, however, gauge freedoms in sequence-function relationships had received little attention. A diffuse body of work had discussed specific approaches for handling gauge freedoms in specific modeling contexts, e.g., in additive models for transcription factor binding specificity [Kinney et al., 2007, Stormo, 2011, Rube et al., 2022], pairwise models of protein fitness landscapes [Weigt et al., 2009, Marks et al., 2011, Ekeberg et al., 2013, Barton et al., 2016, Cocco et al., 2018, Haldane and Levy, 2019, Gerardos et al., 2022, Hsu et al., 2022], and all-order interaction models [Zamuner and Rios, 2021, Feinauer et al., 2022, Feinauer and Borgonovo, 2022]. The issue had also arisen implicitly in the literature concerning alternative parameterizations for genetic interactions [Poelwijk et al., 2019, Metzger et al., 2024, Park et al., 2024b, Dupic et al., 2024, Park et al., 2024a, Faure et al., 2024]. However, a unified understanding of gauge freedoms in sequence-function relationships had yet to be developed.

To address this need, we recently developed a comprehensive mathematical theory of gauge freedoms in generalized one-hot models [Posfai et al., 2025b,a]. In particular, Posfai et al. [2025b] introduced a specific family of gauges that can be imposed through the projection of model parameters via matrix multiplication. These gauges have a simple mathematical form and include nearly all of the gauges used in the prior literature. However, carrying out this matrix operation requires storing and manipulating dense projection matrices, the sizes of which scale quadratically with the number of model parameters, and is impractical for models with more than a few thousand parameters.

Here we introduce GaugeFixer, an open-source Python package that overcomes this computational limitation. By exploiting the mathematical structure of generalized one-hot models and the projection matrices described by Posfai et al. [2025b], GaugeFixer achieves linear scaling in both memory and computation time, enabling rapid gauge fixing for models with millions of parameters. To demonstrate its utility, we apply GaugeFixer here to a fitness landscape for Shine-Dalgarno sequences, represented by a model with nearly 2 million parameters [Kuo et al., 2020, Martí-Gómez et al., 2025]. This analysis illustrates how GaugeFixer can be used to reveal shared sequence requirements across different ribosome binding registers, a phenomenon that is otherwise difficult to discern within such complex sequence-function landscapes.

Background

Generalized one-hot models

We consider generalized one-hot models that predict a scalar function $f(s)$ for sequences s of fixed length L. Of particular interest among these models are all-order models, which describe interactions of every order from 0 through L. Such models can represent any possible sequence-function relationship, but the number of parameters they require grows exponentially with sequence length. It is therefore common to use models that include only interactions up to a specified order (K-order models) or only between nearby positions (nearest-neighbor, or K-adjacent models). These are examples of hierarchical models, a class of generalized one-hot models that remain tractable for longer sequences while still capturing important biological dependencies. See Methods for more information.

Minimal gauge fixing example

We now illustrate the process of gauge fixing using a minimal example of a sequence-function landscape. Consider sequences of length $L = 1$ built using a two-character alphabet $\{\mathrm{A}, \mathrm{B}\}$. There are only two possible sequences in this example: $s = \mathrm{A}$ and $s = \mathrm{B}$. Consequently, the landscape $f(s)$ has only two degrees of freedom. The corresponding all-order model, however, has three parameters—a constant-effect parameter (θ_0) and two additive parameters (θ_1^{A} and θ_1^{B}) that describe the effect of either possible character at position 1 in the sequence. The all-order model therefore exhibits one gauge freedom. Fixing the gauge involves introducing an additional constraint that removes this degree of freedom, leading each valid choice of parameters to produce a unique sequence-function landscape.

Figure 1A illustrates this example. The orange dot shows one choice of parameter vector, $\vec{\theta} = [\theta_0, \theta_1^{\mathrm{A}}, \theta_1^{B}] = [0, 0, 2]$. This choice yields the following sequence-function landscape: $f(A) = \theta_0 + \theta_1^{\mathrm{A}} = 0$, $f(B) = \theta_0 + \theta_1^{\mathrm{B}} = 2$. There are, however, an infinite number of equivalent parameter vectors that yield this same landscape, namely vectors of the form $[z, -z, 2 - z]$ where z is any real number (red line). Fixing the gauge involves specifying an additional constraint that selects a single point on this line. The blue plane in Figure 1A represents one such constraint: the zero-sum gauge, which requires that the additive parameters sum to zero at each position ($\theta_1^{\mathrm{A}} + \theta_1^{\mathrm{B}} = 0$). The resulting gauge-fixed parameters (orange dot) lie at the intersection of the line of equivalent parameters with the zero-sum plane ($\vec{\theta}_{\text{fixed}} = [1, -1, 1]$; green dot). Note that no choice of parameters in the zero-sum gauge other than $\vec{\theta}_{\text{fixed}}$ will yield the specific landscape $f(A) = 0$, $f(B) = 2$.

Families of gauges

In previous work [Posfai et al., 2025b], we introduced a family of gauges for the all-order interaction model. These gauges are parameterized by two quantities: a non-negative number λ and a probability distribution π over sequences. λ controls how much explanatory power is allocated across interaction orders, while π specifies the distribution of sequences against which parameter effects are expressed. This λ, π family of gauges has multiple attractive properties. In particular, it encompasses the most commonly used gauges in the literature including the trivial gauge, the euclidean gauge, the equitable gauge, the zero-sum gauge, and the wild-type gauge. See Posfai et al. [2025b] for details.

A particularly useful subset of gauges in this family are the hierarchical gauges, which are obtained in the limit where λ approaches infinity. In these gauges, lower-order terms explain as much of the landscape's variance as possible, while higher-order interaction terms capture only residual variation that cannot be attributed to lower-order effects. Unlike other gauges in the λ, π family, hierarchical gauges can be applied to all hierarchical models. The values of model parameters in hierarchical gauges also have a natural interpretation: each parameter represents the average effect of introducing specific characters at specific positions, compared to the effect expected from lower-order terms, when sequences are drawn from the distribution π. By choosing different distributions π, one can observe how sequence-function relationships vary across different regions of sequence space. For example, a uniform distribution yields global averages, whereas distributions concentrated near particular sequences reveal local landscape structure.

Results

The GaugeFixer algorithm

Fixing the gauge requires projecting parameter vectors onto lower-dimensional subspaces. The projection matrices for gauges in the λ, π family have a special structure: they can be written as Kronecker products of L much smaller matrices, one matrix for each sequence position [Posfai et al., 2025b]. GaugeFixer exploits this factorization to compute projections without ever constructing the full projection matrix. This is possible because matrix-vector products involving a Kronecker product of L matrices can be computed iteratively, applying one small matrix at a time to successive dimensions of the parameter array [Martí-Gómez et al., 2025]. This reduces both the memory requirements and computation time from $\mathrm{O}(M^2)$ to $\mathrm{O}(M)$, where M is the number of model parameters. For hierarchical models more generally, GaugeFixer decomposes the model into a sum of all-order models restricted to subsets of positions, applies the efficient projection algorithm to each, and sums the results together. See Methods for details.

Figures 1B,C show the performance advantage of this approach. Compared to direct matrix multiplication, GaugeFixer achieves orders of magnitude improvement in both runtime and memory usage. This makes it possible to gauge-fix models with millions of parameters in just a few seconds on a standard laptop computer. By contrast, gauge-fixing becomes impractical beyond about 10^4 parameters when using explicit projection matrices.

Analysis of the Shine-Dalgarno fitness landscape

To illustrate the utility of GaugeFixer, we analyzed the fitness landscape of the Shine-Dalgarno (SD) sequence, a motif in bacterial messenger RNA that facilitates translation initiation by base-pairing with the 3' tail of the 16S ribosomal RNA [Shine and Dalgarno, 1975]. Kuo et al. [2020] previously measured the translational activity of nearly every possible 9-nucleotide RNA sequence. Subsequent modeling efforts inferred a sequence-function landscape from this data, corresponding to an all-order model with $5^9 = 1,953,125$ parameters [Zhou et al., 2022, Martí-Gómez et al., 2025]. This landscape contains multiple fitness peaks corresponding to the canonical AGGAG motif positioned in different registers relative to the start codon.

We used GaugeFixer to quantitatively characterize the local structure around fitness peaks in the landscape inferred by Martí-Gómez et al. [2025]. For each peak we defined a distribution π that fixes the AGGAG motif in the corresponding register while randomly drawing characters at the remaining positions (Figure 1D). We then imposed the hierarchical gauge corresponding to each distribution π and examined the resulting parameter values. In this gauge, the constant, additive, and pairwise parameters can be respectively interpreted as the average phenotype, average single-nucleotide effect, and average epistatic effect observed when mutations are introduced into sequences randomly drawn from π. We emphasize that these different parameter values, corresponding to different gauges, are simply different representations of the same model.

The constant term, θ_0, represents the mean fitness of sequences with AGGAG in the specified register. We find that this value is highest for registers -12 and -11, consistent with known optimal spacing requirements for translation initiation (Figure 1E). In register -9, by contrast, the mean fitness is much lower, indicating markedly lower translation on average.

The additive parameters in the core region, shown in Figure 1F, reveal how individual nucleotide mutations away from the AGGAG motif affect translational efficiency in each register. As expected, these mutations are overwhelmingly deleterious. The effects are also remarkably consistent across registers, though some differences emerge near the boundaries.

The pairwise interaction parameters in the core region, shown in Figure 1G, capture the effects of mutating pairs of nucleotides within the AGGAG motif beyond what additive effects alone predict. These interactions are also remarkably consistent across registers. Predominantly positive values are observed, indicating that combinations of mutations tend to be less deleterious than expected from their individual effects, one hallmark of global epistasis [Otwinowski et al., 2018, Kinney and McCandlish, 2019].

Finally, we compared the constant, additive, and pairwise parameters in the core region across registers (Figure 1H). The results show that neighboring registers tend to have more similar parameters, while more distant registers show greater divergence. This smooth variation suggests that ribosomal binding preferences change gradually as a function of distance from the start codon.

Discussion

Here we introduced GaugeFixer, a Python package that implements a computationally efficient algorithm for removing gauge freedoms from generalized one-hot models of sequence-function relationships. By exploiting the Kronecker factorization of projection operators and the specific mathematical structure of the models, GaugeFixer dramatically reduces runtime and memory requirements, thus enabling rapid gauge fixing for models with millions of parameters.

We wish to emphasize that gauge fixing is fundamentally different from parameter inference. Both procedures are an essential part of modeling sequence-function relationships, but they serve orthogonal purposes. Inference is the process of identifying parameters that make a model's predictions best fit one's data, but without regard to parameter interpretation. Gauge-fixing, by contrast, alters model parameters in ways that have no effect on model predictions, but which are important for the interpretation of those parameters. GaugeFixer makes this distinction explicit by providing utilities to convert precomputed parameter vectors into any chosen gauge. A related but more subtle point is

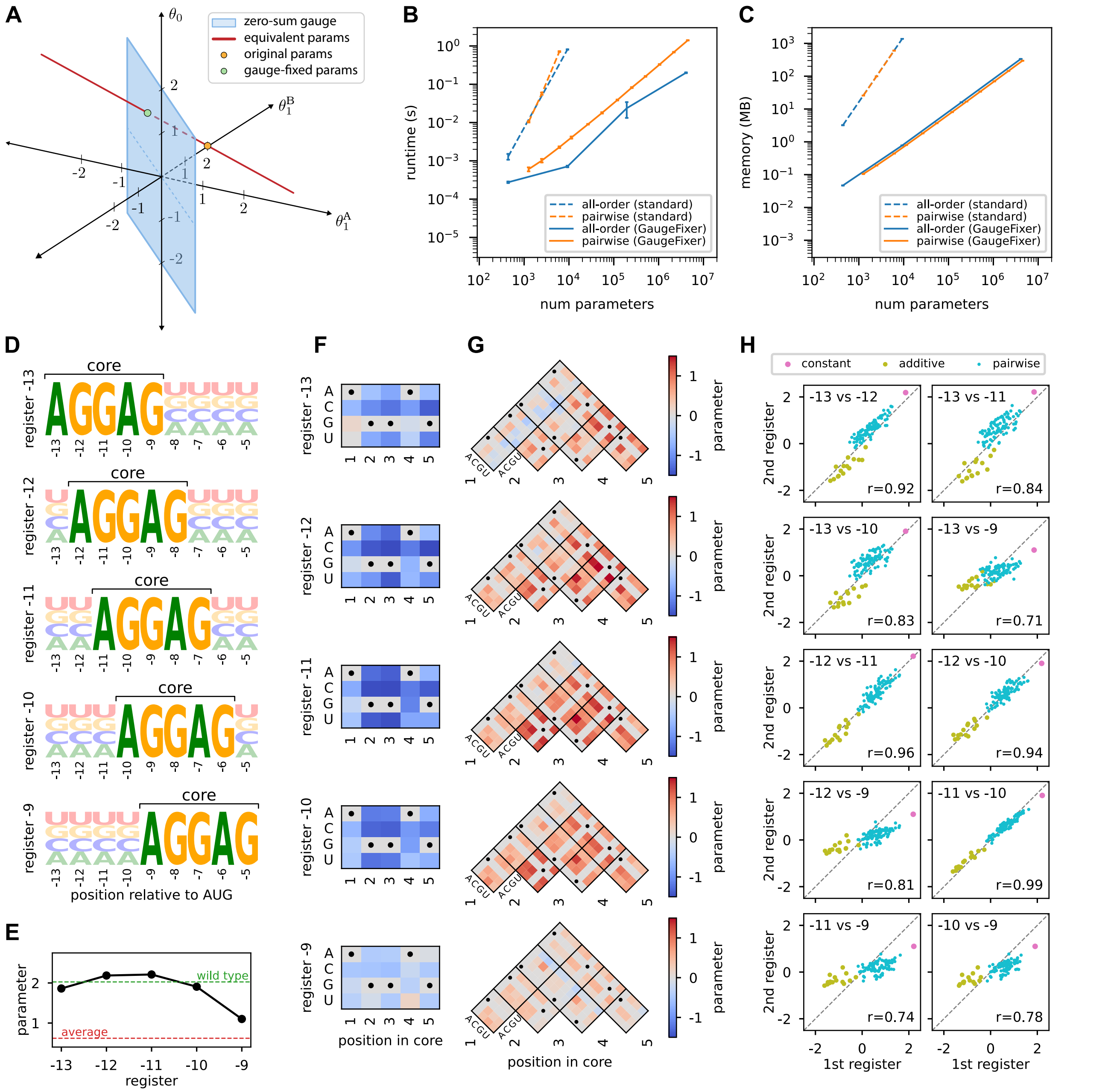

Figure 1: Illustration of gauge-fixing and demonstration of GaugeFixer. (A) Minimal example of gauge fixing. Shown is the space of all possible three-dimensional parameter vectors, $\vec{\theta} = [\theta_0, \theta_1^{\rm A}, \theta_1^{\rm B}]$, for the all-order interaction model on sequences of length $L = 1$ built from the alphabet $\{A, B\}$. Orange dot: initial choice of $\vec{\theta} = [0, 0, 2]$, which yields $f(A) = 0$, $f(B) = 2$. Red line: parameter vectors equivalent to $\vec{\theta}$. Blue plane: parameter vectors in the zero-sum gauge, i.e., that satisfy $\theta_1^{\rm A} + \theta_1^{\rm B} = 0$. Green dot: gauge-fixed parameters $\vec{\theta}_{\rm fixed} = [1, -1, 1]$. (B,C) Algorithm performance when fixing the gauge of all-order and pairwise protein models with increasing number of parameters. Shown are the runtime (B) and peak memory (C) requirements for GaugeFixer and for standard matrix multiplication. Error bars represent the standard deviation in runtime and memory usage over 10 projections of parameter vectors drawn i.i.d. from a normal distribution. (D) Sequence logos showing the probability distributions π corresponding to different peaks in the Shine-Dalgarno fitness landscape measured by Kuo et al. [2020] and modeled by Martí-Gómez et al. [2025]. Each distribution has a fixed AGGAG motif at a specific register relative to the start codon. (E,F,G) Gauge-fixed constant (E), additive (F), and pairwise (G) parameters in the hierarchical gauge associated with each register. Horizontal dashed lines in (E) represent the average phenotype across all possible sequences (red) and the phenotype of the wild-type sequence AAGGAGGUG (green), which occurs in the 5'UTR of the *dmsC* gene of *Escherichia coli*. (H) Scatterplots comparing the values of gauge-fixed parameters observed in the core region at different registers.

that some regularization schemes and/or prior distributions used during inference yield parameters in specific gauges [Posfai et al., 2025b, Petti et al., 2025]. This does not, however, eliminate the need for post-inference gauge fixing. On the contrary, the most common/effective regularization methods often produce parameters in gauges that are not consistent with how one wishes to interpret those parameters.

Although GaugeFixer is designed to work with specific classes of linear models (all-order models and hierarchical models), it can also be applied to nonlinear and nonparametric models, such as neural networks or Gaussian processes. This is done by representing (or approximating) the model-predicted landscape using an all-order (or hierarchical) model, then applying GaugeFixer to the parameters of that model. Here we demonstrated this capability by analyzing a Shine-Dalgarno fitness landscape modeled using Gaussian process regression [Zhou et al., 2022, Martí-Gómez et al., 2025]. Martí-Gómez et al. [2025] recently described gpmap-tools, a Python package that enables such Gaussian process modeling when sequences are short enough to enumerate all possible interaction terms. GaugeFixer complements this functionality by enabling post-hoc computation of gauge-fixed parameters for any of these models. We note, however, that recent theoretical results suggest the possibility of directly computing the posterior distribution over specific gauge-fixed parameters using Gaussian processes defined over sequences of arbitrary length, thus bypassing the need to first construct explicit all-order models [Petti et al., 2025].

Different gauges have different advantages when interpreting model parameters. For instance, the influence of global epistasis is more evident when parameters are represented in the wild-type gauge, whereas using additive models to illustrate the specificities of transcription factors, or computing contact maps in pairwise interaction models, typically requires the models to be in the zero-sum gauge [Kinney et al., 2007, Stormo, 2011, Rube et al., 2022, Marks et al., 2011, Ekeberg et al., 2013]. Gauge-fixing is conceptually related to other techniques for interpretation of neural network models such as global importance analysis [Koo et al., 2021], which quantifies the average effect of placing a specific sequence pattern, e.g., a transcription factor binding site, within random genetic backgrounds. While this method has the advantage that it can be applied to any model, gauge-fixing enables exact computation, not only of the average effect of a sequence pattern over random genetic backgrounds, but also the expected effects of mutations and their combinations. By doing so, gauge-fixing facilitates the investigation of how mutational effects and their interactions change across different regions of sequence space, providing an intuitive way to interpret the higher-order interactions that make them different.

GaugeFixer thus provides a high-performance software library for fixing the gauge of commonly used sequence-to-function models, filling an important gap in the computational tools available for interpreting sequence-function relationships.

Methods

Generalized one-hot models

We consider scalar-valued sequence-function relationships $f(s)$ where s is a sequence of fixed length L built from an alphabet $\mathcal{A} = \{c_1, c_2, \ldots, c_\alpha\}$ comprising α distinct characters.[1] Let $S = \{1, 2, \ldots, L\}$ denote the set of positions within s. Linear models of sequence-function relationships have the form

$$f(s) = \vec{\theta} \cdot \vec{x}(s), \tag{1}$$

where $\vec{\theta}$ is a vector of model parameters and $\vec{x}(s)$ is a vector of sequence-dependent features.

This work focuses on a specific class of linear models, the generalized one-hot models described by Posfai et al. [2025b,a]. In these models, the sequence-dependent features are binary indicator functions over subsequences. Formally, each feature is given by

$$x_U^u(s) = \begin{cases} 1 & \text{if } s[U] = u \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $U \subseteq S$ is a subset of positions, $s[U]$ indicates the subsequence of s that occurs at these positions, and u is a sequence of length $|U|$ (i.e., $u \in \mathcal{A}^{|U|}$).[2] Generalized one-hot models also require that $\vec{x}(s)$ be equivariant under position-specific character permutations [Posfai et al., 2025a]. This means that if x_U^u occurs in $\vec{x}$ for some U and subsequence u, so must $x_U^{u'}$ for every possible subsequence u' of size $|U|$. The set of features in a generalized one-hot model is therefore defined by an alphabet $\mathcal{A}$ and a collection of position sets $V = \{U_1, U_2, \ldots, U_n\}$ via

$$f(s) = \sum_{U \in V} \sum_{u \in \mathcal{A}^{|U|}} \theta_U^u x_U^u(s). \tag{3}$$

All-order models and hierarchical models

All-order models are generalized one-hot models that include features for all possible subsequences at all possible sets of positions. This means that V includes every possible position set $U \subseteq S$. V is therefore the powerset of S, denoted $V = \mathcal{P}(S)$. Because the subsets of S fully define V, we say that S is the generating position set of V.

Hierarchical models are more general than all-order models. Each hierarchical model is defined by one or more generating position sets $S' \subseteq S$, the subsets of which together comprise V. Formally, if $W = \{S'_1, S'_2, \ldots, S'_m\}$ denotes the set of generating position sets, then $V = \bigcup_{S' \in W} \mathcal{P}(S')$ is the union of the powersets of these generating position sets. Common examples of hierarchical models include additive models (defined by all position sets of size 0 and 1), pairwise models (all position sets up to size 2), nearest-neighbor models (same as pairwise but with size-2 position sets restricted to adjacent positions), K-order models (all position sets up to size K), and K-adjacent models (all position sets up to size K but restricted K adjacent positions).[3]

Gauge-fixing projection matrices

The λ, π family is a set of gauges for the all-order interaction model. It is parameterized by a non-negative number λ and a probability

[1] GaugeFixer actually supports sequences built using different alphabets at different positions, with no need for these alphabets to be of the same size. The formulas in this section generalize in a straightforward way to accommodate this more general use case.

[2] Sets of positions are called "orbits" in [Posfai et al., 2025b,a]

[3] Posfai et al. [2025a] adopted a different definition of K-adjacent models, one that makes these models non-hierarchical. We favor the present definition, which is hierarchical and thus amenable to gauge fixing using hierarchical gauges.

distribution π over sequences that factorizes over positions. This family includes many commonly used gauges. In particular, the hierarchical gauges correspond to setting $\lambda = \infty$ and generalizes two commonly used gauges. The zero-sum gauge, commonly used when modeling transcription factor binding to DNA, is the hierarchical gauge that results from π being uniform. Similarly, defining π to be an indicator function for a specific sequence yields the wild-type gauge, which is commonly used in studies of protein fitness landscapes.

Fixing the gauge of all-order models is accomplished by linear projection. In this case there are $M = (\alpha + 1)^L$ parameters. Given an M-dimensional vector $\vec{\theta}$ of initial parameter values, one can obtain the gauge-fixed parameters $\vec{\theta}_{\text{fixed}}$ via multiplication with an appropriate $M \times M$ projection matrix P (which depends on λ and π):

$$\vec{\theta}_{\text{fixed}} = P\vec{\theta}. \tag{4}$$

Importantly, the projection matrix can be written as a Kronecker product of position-specific matrices:

$$P = P_1 \otimes P_2 \otimes \cdots \otimes P_L, \tag{5}$$

where each P_l is the $(\alpha + 1) \times (\alpha + 1)$ matrix

$$P_l = \begin{pmatrix} \eta & \pi_l^{c_1}\eta & \pi_l^{c_2}\eta & \cdots & \pi_l^{c_\alpha}\eta \\ 1-\eta & 1-\pi_l^{c_1}\eta & -\pi_l^{c_2}\eta & \cdots & -\pi_l^{c_\alpha}\eta \\ 1-\eta & -\pi_l^{c_1}\eta & 1-\pi_l^{c_2}\eta & \cdots & -\pi_l^{c_\alpha}\eta \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1-\eta & -\pi_l^{c_1}\eta & -\pi_l^{c_2}\eta & \cdots & 1-\pi_l^{c_\alpha}\eta \end{pmatrix}, \tag{6}$$

π_l^c is the probability of character c occurring at position l, and $\eta = \lambda/(1 + \lambda)$. See Posfai et al. [2025b] for details.

Although the λ, π gauges are defined in the context of all-order models, the specific class of hierarchical gauges can also be applied to the more general class of hierarchical models. This is done by treating each hierarchical model on sequences of length L as an all-order model on sequences of the same length that has unused parameters set to zero. This works because hierarchical gauges preserve these sets of zero-valued parameters. Other λ, π gauges do not preserve these zero-valued parameters, and therefore are restricted in their application to all-order models only.

Efficient gauge fixing via Kronecker factorization

To compute $P\vec{\theta}$, we first reshape $\vec{\theta}$ to be an L-dimensional tensor having size $(\alpha + 1)$ along each dimension. The expression in Eq. 4 then becomes

$$[\vec{\theta}_{\text{fixed}}]_{i_1 \cdots i_L} = \sum_{j_1 \cdots j_L} [P_1]_{i_1 j_1} \cdots [P_L]_{i_L j_L} [\vec{\theta}]_{j_1 \cdots j_L}. \tag{7}$$

We compute Eq. 7 iteratively: first we define $\vec{\theta}^{(0)} = \vec{\theta}$; then for $l = 1, 2, \cdots, L$ we compute

$$\vec{\theta}^{(l)}_{i_1 \cdots i_L} = \sum_{j_l} [P_l]_{i_l j_l} [\vec{\theta}^{(l-1)}]_{i_1 \cdots i_{l-1} j_l i_{l+1} \cdots i_L}, \tag{8}$$

which ultimately yields $\vec{\theta}_{\text{fixed}} = \vec{\theta}^{(L)}$. This computation is formalized in Algorithm 1, which uses a single intermediate parameter vector $\vec{\psi}$ to avoid storing all $\vec{\theta}^{(l)}$ simultaneously. Algorithm 1 has a computational complexity of $\mathrm{O}(L(\alpha + 1)M)$. This compares favorably to the computational complexity of direct matrix multiplication of $\vec{\theta}$ by P, which is $\mathrm{O}(M^2)$. The memory requirements of Algorithm 1 also scale linearly with M, as opposed to quadratically with M for direct matrix multiplication.

Algorithm 1 Fixing the gauge of all-order models

```
Require: $\lambda, \pi, \vec{\theta}_{\text{init}}$
$\vec{\psi} \leftarrow$ reshape($\vec{\theta}_{\text{init}}$)
for $l = 1$ to $L$ do
    $P_l \leftarrow$ ProjectionMatrix($\lambda, \pi_l$)        ▷ Equation 6
    $\vec{\psi} \leftarrow$ tensordot($P_l, \vec{\psi}, l$)        ▷ Equation 8
end for
$\vec{\theta}_{\text{fixed}} \leftarrow$ reshape($\vec{\psi}$)
return $\vec{\theta}_{\text{fixed}}$
```

Hierarchical model decomposition

To project a hierarchical model into a hierarchical gauge, we leverage the fact that the parameters of the model can be expressed as a sum of the parameters of multiple all-order models, each restricted to one of the generating position sets. Let $\vec{\theta}$ be the parameter vector of the hierarchical model expressed as an $(\alpha+1)^L$-dimensional all-order parameter vector (with unused parameters set to zero), and let W denote the set of generating position sets. We first decompose $\vec{\theta}$ as

$$\vec{\theta} = \sum_{S' \in W} \vec{\theta}_{S'} \tag{9}$$

where each $\vec{\theta}_{S'}$ is an all-order parameter vector for the position set S', i.e., has nonzero parameters θ_U^u only for $U \subseteq S'$. We note that there are multiple ways to carry out this decomposition, all of which yield the same gauge-fixed parameters. The gauge-fixed parameters are then computed as

$$\vec{\theta}_{\text{fixed}} = \sum_{S' \in W} P\vec{\theta}_{S'} \tag{10}$$

Importantly, each term on the right-hand side can be computed using only the Kronecker factors P_l corresponding to positions $l \in S'$. Thus, there is never any need to represent the full $(\alpha + 1)^L$-dimensional vectors $\vec{\theta}$, $\vec{\theta}_{S'}$, or $\vec{\theta}_{\text{fixed}}$.

Algorithm 2 formalizes this procedure. Here, $\vec{\theta}[S']$ indicates an $(\alpha + 1)^{|S'|}$-dimensional vector composed only of the elements θ_U^u for which $U \subseteq S'$, and $\pi[S']$ represents the marginal distribution π over subsequences on S'. Note that the iterative form of this algorithm allows us to avoid simultaneously storing the nonzero elements of all $\vec{\theta}_{S'}$ vectors.

Algorithm 2 Fixing the gauge of hierarchical models.

```
Require: $\pi, \vec{\theta}_{\text{init}}, W$
$\vec{\theta}_{\text{fixed}} \leftarrow \vec{0}$
$\vec{\theta}_{\text{tmp}} \leftarrow \vec{\theta}_{\text{init}}$
for $S'$ in $W$ do
    $\vec{\omega} \leftarrow \vec{\theta}_{\text{tmp}}[S']$        ▷ Note: $\vec{\omega} = \vec{\theta}_{S'}[S']$
    $\vec{\omega}_{\text{fixed}} \leftarrow$ FixAllOrders($\infty, \pi[S'], \vec{\omega}$)        ▷ Algorithm 1
    $\vec{\theta}_{\text{fixed}}[S'] \leftarrow \vec{\theta}_{\text{fixed}}[S'] + \vec{\omega}_{\text{fixed}}$
    $\vec{\theta}_{\text{tmp}}[S'] \leftarrow \vec{0}$
end for
return $\vec{\theta}_{\text{fixed}}$
```

Competing interests

The authors declare no competing interests.

Author contributions statement

C.M-G. and J.B.K. conceived the project, wrote the package, and performed the research. C.M-G., D.M.M, and J.B.K. wrote the manuscript. D.M.M. and J.B.K. supervised the research.

Acknowledgments

This work was supported by NIH grants R01HG011787 (J.B.K, D.M.M.), R35GM133777 (J.B.K.), and R35GM133613 (D.M.M., C. M-G.). Computational equipment was supported by NIH grant S10OD028632.

References

J P Barton, E De Leonardis, A Coucke, and S Cocco. ACE: adaptive cluster expansion for maximum entropy graphical model inference. *Bioinformatics*, 32(20):3089–3097, 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw328.

Nathan M. Belliveau, Stephanie L. Barnes, William T. Ireland, Daniel L. Jones, Michael J. Sweredoski, Annie Moradian, Sonja Hess, Justin B. Kinney, and Rob Phillips. Systematic approach for dissecting the molecular mechanisms of transcriptional regulation in bacteria. *Proceedings of the National Academy of Sciences*, 115(21):201722055, 05 2018. ISSN 0027-8424. doi: 10.1073/pnas.1722055115.

Simona Cocco, Christoph Feinauer, Matteo Figliuzzi, R Monasson, and Martin Weigt. Inverse statistical physics of protein sequences: a key issues review. *Reports on Progress in Physics*, 81(3):032601, 2018. ISSN 0034-4885. doi: 10.1088/1361-6633/aa9965.

Thomas Dupic, Angela M. Phillips, and Michael M. Desai. Protein sequence landscapes are not so simple: on reference-free versus reference-based inference. *bioRxiv*, page 2024.01.29.577800, 2024. doi: 10.1101/2024.01.29.577800.

Magnus Ekeberg, Cecilia L ovkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E*, 87(1): 012707, 2013. ISSN 1539-3755. doi: 10.1103/physreve.87.012707.

Andre J. Faure and Ben Lehner. MoCHI: neural networks to fit interpretable models and quantify energies, energetic couplings, epistasis, and allostery from deep mutational scanning data. *Genome Biology*, 25(1):303, 2024. ISSN 1474-7596. doi: 10.1186/s13059-024-03444-y.

Andre J. Faure, Júlia Domingo, Jörn M. Schmiedel, Cristina Hidalgo-Carcedo, Guillaume Diss, and Ben Lehner. Mapping the energetic and allosteric landscapes of protein binding domains. *Nature*, 604(7904):175–183, 2022. ISSN 0028-0836. doi: 10.1038/s41586-022-04586-4.

Andre J. Faure, Ben Lehner, Verónica Miró Pina, Claudia Serrano Colome, and Donate Weghorn. An extension of the walsh-hadamard transform to calculate and model epistasis in genetic landscapes of arbitrary shape and complexity. *PLOS Computational Biology*, 20(5):e1012132, 2024. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1012132.

Christoph Feinauer and Emanuele Borgonovo. Mean Dimension of Generative Models for Protein Sequences. *bioRxiv*, page 2022.12.12.520028, 2022. doi: 10.1101/2022.12.12.520028.

Christoph Feinauer, Barthelemy Meynard-Piganeau, and Carlo Lucibello. Interpretable pairwise distillations for generative protein sequence models. *PLoS Computational Biology*, 18(6): e1010219, 2022. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1010219.

Andonis Gerardos, Nicola Dietler, and Anne-Florence Bitbol. Correlations from structure and phylogeny combine constructively in the inference of protein partners from sequences. *PLoS Computational Biology*, 18(5):e1010147, 2022. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1010147.

Allan Haldane and Ronald M. Levy. Influence of multiple-sequence-alignment depth on Potts statistical models of protein covariation. *Physical Review E*, 99(3):032405, 2019. ISSN 2470-0045. doi: 10.1103/physreve.99.032405.

Thomas A Hopf, John B Ingraham, Frank J Poelwijk, Charlotta P I Schärfe, Michael Springer, Chris Sander, and Debora S Marks. Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135, February 2017. ISSN 1087-0156, 1546-1696. doi: 10.1038/nbt.3769.

Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7):1114–1122, 2022. ISSN 1087-0156. doi: 10.1038/s41587-021-01146-5.

J. D. Jackson and L. B. Okun. Historical roots of gauge invariance. *Reviews of Modern Physics*, 73(3):663–680, 2001. ISSN 0034-6861. doi: 10.1103/revmodphys.73.663.

Justin B Kinney and David M McCandlish. Massively parallel assays and quantitative sequence-function relationships. *Annual Review of Genomics and Human Genetics*, 20(1):99–127, 08 2019. ISSN 1527-8204. doi: 10.1146/annurev-genom-083118-014845.

Justin B. Kinney, Gasper Tkacik, and Curtis G. Callan. Precise physical models of protein-DNA interaction from high-throughput data. *Proceedings of the National Academy of Sciences*, 104(2):501–506, 01 2007. ISSN 0027-8424. doi: 10.1073/pnas.0609908104.

Justin B. Kinney, Anand Murugan, Curtis G. Callan, and Edward C. Cox. Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence. *Proceedings of the National Academy of Sciences*, 107(20): 9158–9163, 05 2010. ISSN 0027-8424. doi: 10.1073/pnas.1004290107.

Peter K. Koo, Antonio Majdandzic, Matthew Ploenzke, Praveen Anand, and Steffan B. Paul. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLOS Computational Biology*, 17(5):e1008925, May 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1008925.

Syue Ting Kuo, Ruey Lin Jahn, Yuan Ju Cheng, Yi Lan Chen, Yun Ju Lee, Florian Hollfelder, Jin Der Wen, and Hsin Hung David Chou. Global fitness landscapes of the Shine-Dalgarno sequence. *Genome Research*, 30(5):711–723, 2020. ISSN 15495469. doi: 10.1101/gr.260182.119.

Debora S. Marks, Lucy J. Colwell, Robert Sheridan, Thomas A. Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3D Structure Computed from Evolutionary Sequence Variation. *PLoS ONE*, 6(12):e28766, 2011. doi: 10.1371/journal.pone.0028766.

Debora S Marks, Thomas A Hopf, and Chris Sander. Protein structure prediction from sequence variation. *Nature Biotechnology*, 30(11):1072–1080, November 2012. ISSN 1087-0156, 1546-1696. doi: 10.1038/nbt.2419.

Carlos Martí-Gómez, Juannan Zhou, Wei-Chia Chen, Justin B. Kinney, and David M. McCandlish. Inference and visualization of complex genotype-phenotype maps with gpmap-tools. *bioRxiv*, page 2025.03.09.642267, 2025. doi: 10.1101/2025.03.09.642267.

Brian PH Metzger, Yeonwoo Park, Tyler N Starr, and Joseph W Thornton. Epistasis facilitates functional evolution in an ancient transcription factor. *Elife*, 12:RP88737, 2024.

Ilaria Mogno, Jamie C Kwasnieski, and Barak A Cohen. Massively parallel synthetic promoter assays reveal the in vivo effects of binding site variants. *Genome Res*, 23(11):1908 – 1915, 2013. ISSN 1088-9051. doi: 10.1101/gr.157891.113.

Jakub Otwinowski, David M McCandlish, and Joshua B Plotkin. Inferring the shape of global epistasis. *Proc Natl Acad Sci USA*, 115(32):E7550 – E7558, 2018. ISSN 0027-8424. doi: 10.1073/pnas.1804015115.

Yeonwoo Park, Brian P.H. Metzger, and Joseph W. Thornton. On the analysis of protein genetic architecture: Response to "protein sequence landscapes are not so simple". *bioRxiv*, 2024a. doi: 10.1101/2024.09.17.613512. URL `https://www.biorxiv.org/content/early/2024/12/06/2024.09.17.613512`.

Yeonwoo Park, Brian PH Metzger, and Joseph W Thornton. The simplicity of protein sequence-function relationships. *Nature Communications*, 15(1):7953, 2024b.

Samantha Petti, Carlos Martí-Gómez, Justin B Kinney, Juannan Zhou, and David M McCandlish. On learning functions over biological sequence space: relating gaussian process priors, regularization, and gauge fixing. *arXiv*, 2025. doi: 10.48550/arxiv.2504.19034.

Frank J Poelwijk, Michael Socolich, and Rama Ranganathan. Learning the pattern of epistasis linking genotype and phenotype in a protein. *Nature Communications*, 10(1):4213, 2019. doi: 10.1038/s41467-019-12130-8.

Anna Posfai, David M. McCandlish, and Justin B. Kinney. Symmetry, gauge freedoms, and the interpretability of sequence-function relationships. *Physical Review Research*, 7(2):023005, 2025a. doi: 10.1103/physrevresearch.7.023005.

Anna Posfai, Juannan Zhou, David M. McCandlish, and Justin B. Kinney. Gauge fixing for sequence-function relationships. *PLOS Computational Biology*, 21(3):e1012818, 2025b. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1012818.

H. Tomas Rube, Chaitanya Rastogi, Siqian Feng, Judith F. Kribelbauer, Allyson Li, Basheer Becerra, Lucas A. N. Melo, Bach Viet Do, Xiaoting Li, Hammaad H. Adam, Neel H. Shah, Richard S. Mann, and Harmen J. Bussemaker. Prediction of protein-ligand binding affinity from sequencing data with interpretable machine learning. *Nature Biotechnology*, 40 (10):1520–1527, 2022. ISSN 1087-0156. doi: 10.1038/s41587-022-01307-0.

J Shine and L Dalgarno. Determinant of cistron specificity in bacterial ribosomes. *Nature*, 254:34–38, 1975.

Gary D Stormo. Maximally efficient modeling of DNA sequence motifs at all levels of complexity. *Genetics*, 187(4):1219 – 1224, 2011. ISSN 0016-6731. doi: 10.1534/genetics.110.126052.

Gary D. Stormo. Modeling the specificity of protein-DNA interactions. *Quantitative Biology*, 1(2):115–130, June 2013. ISSN 2095-4689, 2095-4697. doi: 10.1007/s40484-013-0012-4.

Ammar Tareen, Mahdi Kooshkbaghi, Anna Posfai, William T. Ireland, David M. McCandlish, and Justin B. Kinney. MAVE-NN: learning genotype-phenotype maps from multiplex assays of variant effect. *Genome Biology*, 23(1):98, 2022. ISSN 1474-7596. doi: 10.1186/s13059-022-02661-7.

Martin Weigt, Robert A. White, Hendrik Szurmant, James A. Hoch, and Terence Hwa. Identification of direct residue contacts in protein-protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72, 2009. ISSN 0027-8424. doi: 10.1073/pnas.0805923106.

Gene Yeo and Christopher B Burge. Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. In *Journal of Computational Biology*, volume 11, pages 377–394, March 2004. doi: 10.1089/1066527041410418.

Stefano Zamuner and Paolo De Los Rios. Interpretable Neural Networks based classifiers for categorical inputs. *arXiv*, 2021. doi: 10.48550/arxiv.2102.03202.

Juannan Zhou, Mandy S Wong, Wei-chia Chen, Adrian R Krainer, B Justin, and David M Mccandlish. Higher-order epistasis and phenotypic prediction. *Proc. Natl. Acad. Sci. USA*, 119(39), 2022. doi: https://doi.org/10.1073/pnas.2204233119.

Juannan Zhou, Carlos Martí-Gómez, Samantha Petti, and David M. McCandlish. Learning sequence-function relationships with scalable, interpretable Gaussian processes. *eLife*, 14, November 2025. doi: 10.7554/eLife.108964.1.