bioRχiv, 2025, 1–6
doi:
Original Article

GaugeFixer: Overcoming Non-identifiability in Sequence-to-Function Models

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., 11724, New York, United States

*Corresponding author. jkinney@cshl.edu

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Summary: Computational biology commonly involves the use of mathematical models to describe sequence-function relationships in terms of additive effects and combinations of epistatic effects of varying order. A critical but underappreciated challenge when using such models is that interpreting their parameters requires that all non-identifiable degrees of freedom (called "gauge freedoms") first be eliminated through the introduction of additional mathematical constraints on values (a process called "fixing the gauge"). Recently we introduced a general mathematical theory for how to systematically eliminate gauge freedoms while maintaining biological interpretability. Here we describe GaugeFixer, a Python package that efficiently implements these gauge-fixing methods. By leveraging the Kronecker factorization of projection matrices, GaugeFixer can rapidly process models having millions of parameters, a task that is otherwise computationally onerous. GaugeFixer thus overcomes a major obstacle in the biological interpretation of sequence-function relationships.

Availability and implementation: GaugeFixer can be installed using the pip package manager and is compatible with both Python $\geq$ 3.10. Documentation is provided at http://gaugefixer.readthedocs.io; source code is available at http://github.com/jbkinney/gaugefixer. Code to reproduce the analyses presented here is available at http://github.com/jbkinney/gaugefixer/manuscript

Contact: jkinney@cshl.edu (J.B.K.)

Introduction

Computational biology routinely involves the use of models that describe the quantitative relationship between sequence (e.g. DNA, RNA, or protein sequence) and biological activity. These models have been used to predict the locations of transcription factor binding sites [Stormo, 2013] and splice sites [Yeo and Burge, 2004] along the genome, to predict structural contacts [Marks et al., 2012], to assess the effects of mutations across human proteins Hopf et al. [2017], and to model high-throughput mutagenesis data [Kinney and McCandlish, 2019, Tareen et al., 2022]. Sequence-function models can be represented as linear functions of binary features indicating the presence or absence of particular characters at specific positions in each sequence. The parameters associated to these binary features have interesting mathematical properties and symmetries, such as invariance under permutations of alleles and positions [Posfai et al., 2025b,a]. However, such linear models are overparameterized, resulting in "gauge freedoms", this is, parameter subspaces encoding exactly the same sequence-function map. Thus, before making any scientific claim about the parameter values, one must first remove all gauge freedoms by introducing additional mathematical constraints that provide a unique representation of the model. This process is called "fixing the gauge" [Ekeberg et al., 2013], where "the gauge" refers to the a specific set of constraints and determines the interpretation of model parameters.

In recent work, we have developed a general theory of gauge-freedoms in sequence-function relationships [Posfai et al., 2025b,a]. In this theory, we introduce a parametric family of linear gauges that unifies many of the previously proposed gauges and we proposed a general mathematical strategy for fixing the gauge of a wide range of sequence-function relationships. This strategy uses linear projection matrices to map model parameters to lower-dimensional subspaces of parameter space describing different choices of gauge. While mathematically straight-forward, this method involves computations with projection matrices that scale quadratically with the number of parameters, limiting its practical use to models with only a few thousand parameters.

Here, we introduce a new algorithm for fixing the gauge that scales linearly with the number of parameters by leveraging the mathematical structure of gauge-specific projection matrices to

efficiently project any set of parameter values into a given gauge without explicity building these matrices. We have implemented this algorithm in an open-source and documented Python library called GaugeFixer. To demonstrate its power, we use GaugeFixer to characterize the local sequence dependencies around different peaks in the fitness landscape of the Shine-Dalgarno sequence [Kuo et al., 2020, Martí-Gómez et al., 2025]. In summary, GaugeFixer provides the first standalone and lightweight software tool for fixing the gauge, enabling the biological interpretation of sequence-function relationships at an unprecedented scale.

Gauge freedoms in sequence-function models

In this section, we provide a brief overview of our mathematical formulation of linear models for sequence-function relationships and gauge freedoms presented in recent work [Posfai et al., 2025b,a].

Linear models

We consider the set of sequence-function relationships $f(s)$ where f is a scalar and s is a sequence of fixed lenght L built from a set of α distinct characters $A = \{c_1, c_2, \ldots, c_\alpha\}$ across the set of sites $S = \{1, 2, \ldots, L\}$. In particular, we focus on the set sequence-function model $f(s)$ in this class can be expressed as a linear combination of a predefined set of binary sequence features $\vec{x}(s)$:

$$f(s) = \vec{\theta} \cdot \vec{x}(s) = \sum_{U \in V} \sum_{u} \theta_U^u x_U^u(s), \tag{1}$$

where each feature $x_U^u(s)$ is defined by the presence or absence of a subsequence $u = c_1 c_2 \ldots c_K$ in an "orbit" or subset of sites $U \subseteq S$ at sequence s from a set of n orbits $V = \{U_1, U_2, \ldots, U_n\}$. This particular model representation has the advantage that the values of the parameters are invariant under permutations of positions and alleles, and thus can be interpreted as intrinsic allelic (or features) effects [Posfai et al., 2025a].

Different classes of models in this family are defined by the set of orbits V. For instance, the all-order interaction model is characterized by including every possible orbit of S, known as the power set $V = \mathcal{P}(S) = \{U : U \subseteq S\}$. A more general class of models is the family of hierarchical models, which are defined by a set of m generating orbits $W = \{U_1, U_2, \ldots, U_m\}$, such that they include features for every sub-orbit of any orbit in W given by $V = \bigcup_{U \in W} \mathcal{P}(U)$. For instance, for a model including parameters in the orbit $U = \{1, 2\}$ to be hierarchical, it must also include all parameters in the sub-orbits of U ($\{\{\}, \{1\}, \{2\}\}$). Hierarchical models include the all-order interaction model (for $W = \{S\}$) but also other commonly used models with specific subsets of features e.g. models including features up to Kth order ($W = \{U : |U| = K\}$), such as pairwise ($K = 2$) and 3-way ($K = 3$) interaction models, or models including features defined over orbits of up to K adjacent sites ($W = \{\{l, l+1, \ldots, l+K-1\} \subseteq S\}$), such as nearest-neighbor interaction models ($K = 2$). Whereas the number of parameters to all-order interaction models scale exponentially with sequence length, these other hierarchical models are more practical to describe sequence-function relationships defined over much longer sequences.

Gauge freedoms

Despite the advantages of this family of linear models, these models are overparametrized e.g. all-order interaction models have $M = (\alpha + 1)^L$ parameters but are defined only over α^L distinct sequences of length L. As a result, there is a subspace of parameter space, known as the gauge, encoding any sequence-function model. For instance, we can consider a simple three parameter ($\vec{\theta} = [\theta_0, \theta_1^A, \theta_1^B]$) model defined over sequences of length $L = 1$ with only two alleles $A = \{A, B\}$. If we let $f(A) = 0.5$ and $f(B) = -1$, we can see how there is a linear subspace in parameter space that encodes exactly the same sequence-function map (Figure 1A, red line) e.g. $\vec{\theta} = [0, 0.5, -1]$ and $\vec{\theta} = [0.5, 0, -1.5]$, preventing the direct interpretation of the parameter values.

Thus, it is useful to define gauges, this is, specific parameter subspaces in which parameter values can be interpreted in different ways. For instance, the plane in Figure 1A represents a specific gauge known as the zero-sum gauge, this is, the parameter space in which all the marginals equal zero. In previous work, Posfai et al. [2025b] proposed a parametric family of linear gauges defined by two quantities. The first quantity is a non-negative number λ that controls how much variance in the model's predictions should be explained by the parameters associated to lower-order features. The second quantity, π, is a site-independent probability distribution over sequences used to compute this variance, where π_l^c specifies the probability of character c occuring at position l.

An important subset of the λ, π gauges are defined by $\lambda = \infty$. These gauges are known as hierarchical gauges because parameters in these subspaces maximize the amount of variance (defined according to the distribution π) explained by lower-order features i.e., they maximize the variance of all truncated models formed by removing all terms from f of a given order or greater. This subset of gauges have two important properties. First, they preserve the form of hierarchical models regardless of the choice of π. Second, parameters in any hierarchical gauge can be interpreted as the average effect of introducing a set of characters at specific positions compared with the expected effect from the parameters in the sub-orbit for sequences drawn from given probability distribution. Thus, parameter values in different hierarchical gauges provide information about the intrinsic feature effects across different regions of sequence space defined by π.

Gauge fixing

In order to meaningfully interpret the parameters of a given sequence-function model $\vec{\theta}$, they have to be expressed in a specific λ, π gauge, an operation that we call "fixing the gauge", this is, choosing one among the many possible representations of the model where the parameters have a specific interpretation. In the example shown in Figure 1A, we choose the unique representation of $f(s)$ in the zero-sum gauge subspace, this is, where the two parameter subspaces cross and providing specific interpretations to the parameters: $\theta_0 = -0.25$ represents the average phenotype across the whole landscape, whereas $\theta_1^A = 0.75$ and $\theta_1^B = -0.75$ can be interpreted as the average effect of placing alleles A and B respectively at position 1 across all possible backgrounds.

Posfai et al. [2025b] showed that this can be generally done via linear projection and derived the corresponding projection matrices considering an all-order interaction model:

$$\vec{\theta}_{\text{fixed}} = P\vec{\theta}. \tag{2}$$

In the case of arbitrary hierarchical models, we can treat them as all-order interaction models with unused parameters set to zero. This is often impractical, however, as the number of parameters in the all-order interaction models would be too large to handle e.g. a pairwise defined over a 51 amino acid sequence-function model described in Olson et al. [2014] has a tractable number of parameters $M = 511,021$, but the corresponding all-orders model has an astronomically large number of parameters $M \sim 10^{67}$, rendering this strategy unfeasible in practice. However, as the hierarhical gauges ($\lambda = \infty$) preserve the form of hierarchical models, we can define a projection matrix only for the M non-zero parameters by computing only a small submatrix of P.

In summary, while mathematically simple, this gauge-fixing projection is not trivial to compute in many real-world scenarios, as both the memory costs to store P and the computational cost of computing $P\vec{\theta}$ are $O(M^2)$, limiting its practical use to models with only a few thousand parameters.

Results

An efficient algorithm for fixing the gauge in all-order models

Here, we present a new algorithm for projecting the parameters of any hierarhical linear model into a specific gauge with $O(M)$ memory and computational cost. To do so, we leverage the specific mathematical form of P, defined as tensor products of L distinct subspaces of dimension $\alpha + 1$, one subspace for each position l. Consequently, the gauge-fixing projection matrix can be written as a Kronecker product of L matrices:

$$P = P_1 \otimes P_2 \otimes \cdots \otimes P_L, \tag{3}$$

where P_k is a $(\alpha+1) \times (\alpha+1)$ position-specific projection matrix for any $k \in S$ as defined in Posfai et al. [2025b]. Importantly, the product $P\vec{\theta}$ can be computed recursively using only one P_k at a time [Martí-Gómez et al., 2025] (See Algorithm 1). We first reshape $\vec{\theta}$ to be an L-dimensional tensor having size $(\alpha+1)$ along each dimension. The expression in Eq. 2 then becomes

$$[\vec{\theta}_{\text{fixed}}]_{i_1 \cdots i_L} = \sum_{j_1 \cdots j_L} [P_1]_{i_1 j_1} \cdots [P_L]_{i_L j_L} [\vec{\theta}]_{j_1 \cdots j_L} \tag{4}$$

where each i and j index $0, 1, \ldots, \alpha$. This can be computed by first defining $\vec{\theta}^{(0)} = \vec{\theta}$, then for $k = 1, 2, \cdots, L$ recursively computing

$$\vec{\theta}^{(k)}_{i_1 \cdots i_L} = \sum_{j_k} [P_k]_{i_k j_k} [\vec{\theta}^{(k-1)}]_{i_1 \cdots i_{k-1} j_k i_{k+1} \cdots i_L}, \tag{5}$$

where $\vec{\theta}^{(L)}$ corresponds to the gauge-fixed parameter vector $\vec{\theta}_{\text{fixed}}$.

Algorithm 1 Fixing the gauge of all-order models

Require: $\lambda, \pi, \vec{\theta}$
$\vec{\theta}^{(k-1)} \leftarrow \text{reshape}(\vec{\theta})$
for $k = 1$ in 1 **do**
  $P_k \leftarrow \text{ProjectionMatrix}(\lambda, \pi_k)$ ▷ Posfai et al. [2025b]
  $\vec{\theta}^{(k)} \leftarrow \text{tensordot}(P_k, \vec{\theta}^{(k-1)}, k)$ ▷ Equation 5
  $\vec{\theta}^{(k-1)} \leftarrow \vec{\theta}^{(k)}$
end for
$\vec{\theta}_{\text{fixed}} \leftarrow \text{reshape}(\vec{\theta}^{(k)})$
return $\vec{\theta}_{\text{fixed}}$

Note that equation 5 can be computed in practice as a tensor dot product of the P_k matrix with the tensor $\vec{\theta}^{(k-1)}$. Thus, computation has complexity $O(L(\alpha+1)M)$, as opposed to $O(M^2)$ for the naive approach and memory requirements scale only linearly with the number of parameters M.

Fixing the gauge in hierarchical models

In this section, we present an extended algorithm that allows projecting the parameters of an arbitrary hierarchical model into a specific hierarhical gauge defined by π. We note that any hierarchical model can be decomposed as the sum of m all-order interaction models defined over each of the generating orbits $U' \in W$ defined by $\vec{\theta}_{U'}$:

$$\theta^u_U = \sum_{U' \in W : U \subseteq U'} [\vec{\theta}_{U'}]^u_U. \tag{6}$$

While this decomposition is not unique i.e. there are more parameters in the m all-order models than in the hierarchical model, if the m all-order models are expressed in a given gauge, any linear combination of them will also be in that gauge. Thus, we can take an arbitrary decomposition of $\vec{\theta}$ into the $\vec{\theta}_{U'}$, fix the gauge of the $\vec{\theta}_{U'}$ parameters using the Kronecker structure of the projection matrix $P_{U'}$ for the all-order model defined on each orbit U' and add them up to obtain $\vec{\theta}_{\text{fixed}}$ using equation 6. In practice, we do this via the following Algorithm 2 to avoid storing all $\vec{\theta}_{U'}$ simultaneously.

Algorithm 2 Fixing the gauge of hierarchical models

Require: $\pi, \vec{\theta}, W$
$\vec{\theta}_{\text{source}} \leftarrow \vec{\theta}$
$\vec{\theta}_{\text{fixed}} \leftarrow \vec{0}$
for U' in W **do**
  $\vec{\theta}_{U'} \leftarrow \vec{\theta}_{\text{source}}[U']$
  $[\vec{\theta}_{U'}]_{\text{fixed}} \leftarrow \text{FixAllOrder}(\infty, \pi[U'], \vec{\theta}_{U'})$ ▷ Algorithm 1
  $\vec{\theta}_{\text{fixed}}[U'] \leftarrow \vec{\theta}_{\text{fixed}}[U'] + [\vec{\theta}_{U'}]_{\text{fixed}}$
  $\vec{\theta}_{\text{source}}[U'] \leftarrow \vec{0}$
end for
return $\vec{\theta}_{\text{fixed}}$

GaugeFixer: a software library for fixing the gauge in sequence-function models

We have implemented these computationally efficient algorithms in a lightweight Python library called GaugeFixer. GaugeFixer provides a simple object-oriented interface, where users can define different classes of hierarchical models i.e. all-order, K-order, and K-adjancent interaction models, and project any set of previously inferred parameters into any user-defined gauge λ, π gauge for interpretation of their values.

To evaluate the performance of GaugeFixer's implementation, we computed the running time and memory requirements for fixing the gauge in different types of hierarhical models with increasing number of randomly initialized parameters (Figure 1B and C). GaugeFixer shows over an order of magnitude increase in performance over the naive approach using a dense projection matrix and much better scaling with the number of parameters.

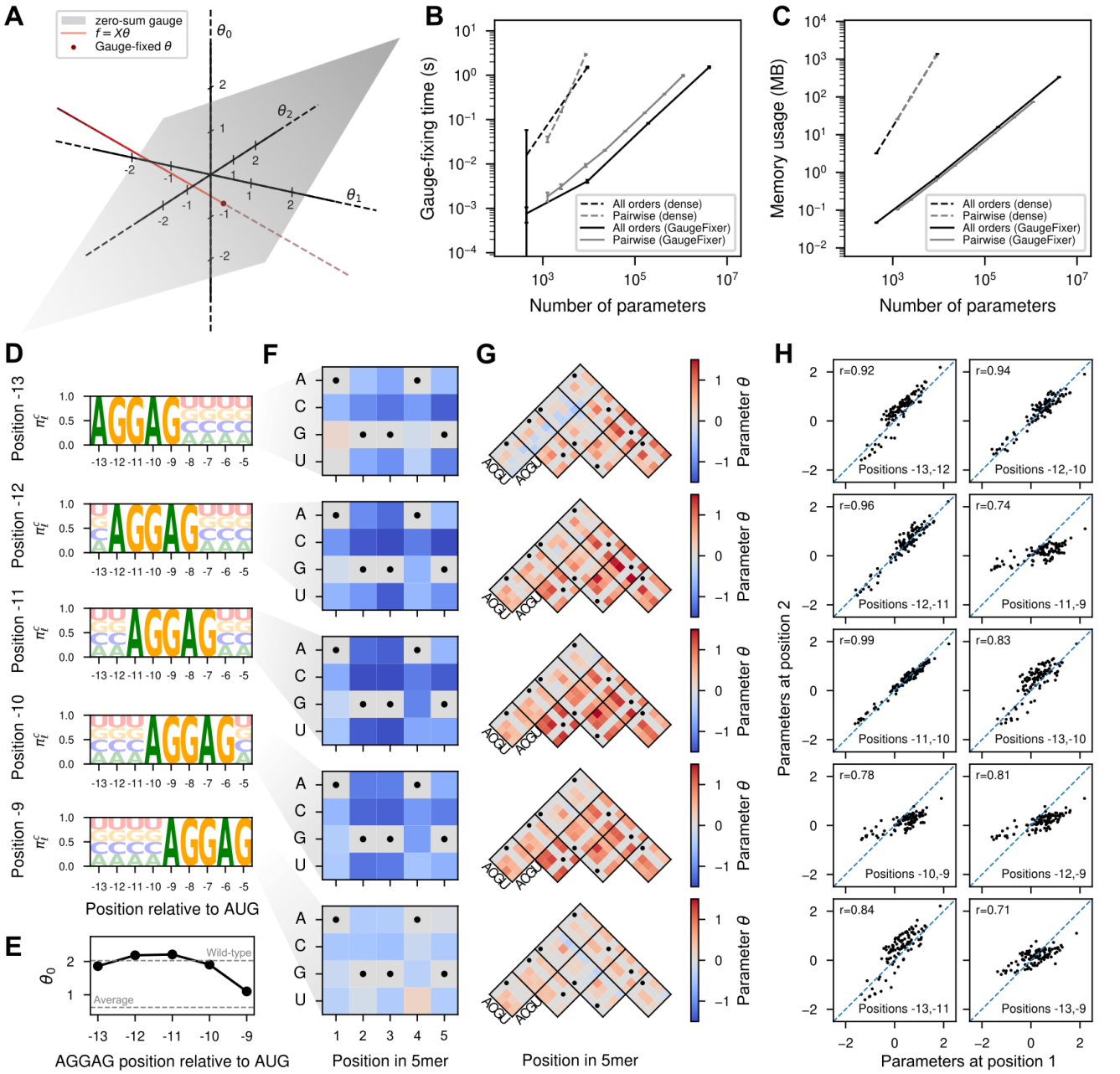

Fig. 1: GaugeFixer: a software package for fixing the gauge in sequence-function models with applications to the Shine-Dalgarno (SD) fitness landscape. (A) Representation of the parameter subspaces for a sequence-function model with one site and two alleles. The red line represents the parameter subspace able to represent the phenotypes [-0.5, 1]. The plan represents the subspace of parameters in the zero-sum gauge. The two subspaces intersect at a unique point, where the model parameters are expressed in the zero-sum gauge. (B,C) Performance of GaugeFixer for fixing the gauge in all orders and pairwise interaction models in protein spaces with increasing number of parameters in terms of running time (B) and memory requirements (C), compared to a naive approach using dense projection matrices. (D) Sequence logos representing the site-independent probability distributions that represent the different peaks of the SD sequence with the AGGAG motif at different positions relative to the start codon. (E,F,G) Representation of the gauge-fixed constant (E), additive (E) and pairwise (F) parameters, respectively, under the hierarchical gauge associated to each probability distribution associated to the AGGAG motif at different positions relative to the start codon AUG. Horizontal dashed lines in (E) represent the average phenotype across all possible sequences and the phenotype of the wild-type sequence AAGGAGGUG in the dmsC 5'UTR. (H) Scatterplots comparing the values of the gauge-fixed parameters of the local pairwise interaction model around each of the fitness peaks defined by the probability distributions in (D).

Analysis of the fitness landscape of the Shine-Dalgarno sequence

To demonstrate the utility of GaugeFixer, we analyze the fitness landscape of the Shine-Dalgarno (SD) sequence, a 5' UTR motif that is critical for translation initiation in prokaryotes [Shine and Dalgarno, 1975]. Using high-throughput fluorescence measurements of nearly every 9-mer [Kuo et al., 2020], we previously inferred an all-order sequence-function model using variance component regression [Zhou et al., 2022] and showed that the landscape contains multiple peaks corresponding to recognition of the canonical motif at different distances from the start codon [Martí-Gómez et al., 2025]. Here, we use GaugeFixer to probe the local structure of each peak in more detail.

In order to define regions of sequence space corresponding to the different fitness peaks, we define a series of site-independent probability distributions π with the AGGAG motif at different positions relative to the start codon. Thus, for a given position p, we set $\pi_p^A = 1$, $\pi_{p+1}^G = 1$, $\pi_{p+2}^G = 1$, $\pi_{p+3}^A = 1$, $\pi_{p+4}^G = 1$, with all other sites drawn uniformly ($\pi_l^c = 0.25$ for $l \notin [p, p+4]$, Figure 1A). For each distribution, we fix the gauge using the hierarchical gauge ($\lambda = \infty$) to concentrate as much explanatory variance as possible into lower-order terms, and then interpret the resulting parameters.

The zeroth-order parameter θ_0 represents the mean phenotype under each distribution (i.e., the average fluorescence intensity of sequences with the AGGAG motif at the specified position). We find that θ_0 is largest when the motif is at positions -12 and -11, consistent with an optimal SD positioning for translation initiation (Figure 1E). In contrast, θ_0 is much closer to the average intensity across all possible sequences for AGGAG motifs located at position -9, suggesting much weaker activation when binding at this position. The parameters θ_l^c associated with the alleles at each position l represent the average effect of fixing a allele c when sequences are drawn from each probability distribution π. Our gauge-fixed θ_l^c show that the average allelic effects within the AGGAG motif are very similar across the different positions (Figure 1F). These allelic effects are slightly different at position -13, where G is preferred at the first position of the 5-mer on average, and at position -9, where U is preferred over A at the fourth position. These differences could be driven by the presence of fixed alleles at positions -14 and -4 adjacent to the randomized motif, whereas the alleles at sites flanking the AGGAG motif at positions -12 to -10 are sampled uniformly. Similarly, the parameters $\theta_{l_1 l_2}^{c_1 c_2}$ associated with pairs of alleles reflect the average deviation from the additive model when pairs of alleles are fixed in sequences drawn from the corresponding probability distributions. These parameters also show consistent patterns of interaction across the different positions and alleles (Figure 1G), further supporting the similarity requirements around each of the peaks. These $\theta_{l_1 l_2}^{c_1 c_2}$ take mostly positive values, suggesting that combinations of mutations tend to have higher function than expected by their average single point mutational effects.

These constant, additive and pairwise interaction terms define a local pairwise interaction model for the sequence requirements at each binding register. Figure 1H quantitatively compares the parameters of these models across every pair of binding positions. These results show that the local sequence-function models around each peak are more similar between neighboring positions, and differ as they become far appart in the primary sequence, suggesting that binding preferences change smoothly as a function of the distance to the start codon. While the parameters of the local model at position -9 are in general of smaller magnitude, they remain substantially correlated with the model parameters for the other positions (Figure 1H).

In summary, GaugeFixer allowed us to explore and compare the structure and sequence determinants of the different peaks in the Shine-Dalgarno fitness landscape, uncovering similarities and differences in the recognition modes at different positions relative to the start codon.

Discussion

Here we introduced GaugeFixer, a lightweight Python library that implements a computationally efficient algorithm for removing unconstrained degrees of freedom ("fix the gauge") from the parameters of a wide range of linear sequence-to-function models. By exploiting the Kronecker factorization of projection operators and the structure of hierarchical models, our new algorithm reduces the time and memory complexity of gauge fixing from $O(M^2)$ to $O(M)$, making it practical for models with millions of parameters.

It is important to emphasize that gauge fixing is completely independent from parameter inference. Although particular regularization schemes or prior distributions can implicitly pick out a gauge during model fitting [Posfai et al., 2025b, Petti et al., 2025], fixing the gauge is a separate post-processing step applied to an already-specified model. GaugeFixer makes this distinction explicit by providing utilities to convert any precomputed parameter vector into a series of different gauges for interpretation.

Gauge fixing can in principled be applied for interpretation of nonlinear and nonparametric models, such as neural networks or Gaussian process models, by representing them as all-order linear models. In this work we used GaugeFixer to extract gauge-fixed parameters from a Shine-Dalgarno sequence-function model inferred using Gaussian process regression [Zhou et al., 2022, Martí-Gómez et al., 2025] allowing us to zoom in into the local sequence requirements in different regions of sequence space. This feasible when sequences are short enough to enumerate the parameters associated to features of every possible order. However, recent theoretical results [Petti et al., 2025] further show that gauge-fixed parameters can be computed for a broad class of Gaussian process models with site-wise product kernels defined over sequences of arbitrary length, opening a practical route to interpret these powerful nonparametric models.

In summary, GaugeFixer provides the first softare library for fixing the gauge of linear models, filling an important gap in the set of computational tools for interpreting models of sequence-function relationships.

Competing interests

The authors declare no competing interests.

Author contributions statement

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

Acknowledgments

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

References

Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 87(1):1–16, 2013. ISSN 15393755. doi: 10.1103/PhysRevE.87.012707. arXiv: 1211.1281.

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. URL `https://www.nature.com/articles/nbt.3769`.

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. Wrote.

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, 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. URL `http://www.nature.com/articles/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.

C Anders Olson, Nicholas C Wu, and Ren Sun. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Current biology : CB*, 24(22):2643 – 2651, 11 2014. ISSN 0960-9822. doi: 10.1016/j.cub.2014.09.072.

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.

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.

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

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. URL `https://onlinelibrary.wiley.com/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.

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. URL `http://www.liebertpub.com/doi/10.1089/1066527041410418`. Issue: 2-3 ISSN: 10665277.

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.