The Annals of Applied Statistics
2019, Vol. 13, No. 2, 1268–1294
https://doi.org/10.1214/18-AOAS1233

SURVIVAL ANALYSIS OF DNA MUTATION MOTIFS WITH PENALIZED PROPORTIONAL HAZARDS

BY JEAN FENG[*,1,6], DAVID A. SHAW[†,3,4,6], VLADIMIR N. MININ[‡,3,7], NOAH SIMON[*,2,7] AND FREDERICK A. MATSEN IV[†,3,4,5,7]

University of Washington[*]*, Fred Hutchinson Cancer Research Center*[†] *and University of California, Irvine*[‡]

Antibodies, an essential part of our immune system, develop through an intricate process to bind a wide array of pathogens. This process involves randomly mutating DNA sequences encoding these antibodies to find variants with improved binding, though mutations are not distributed uniformly across sequence sites. Immunologists observe this nonuniformity to be consistent with "mutation motifs" which are short DNA subsequences that affect how likely a given site is to experience a mutation. Quantifying the effect of motifs on mutation rates is challenging. A large number of possible motifs makes this statistical problem high dimensional, while the unobserved history of the mutation process leads to a nontrivial missing data problem. We introduce an ℓ_1-penalized proportional hazards model to infer mutation motifs and their effects. In order to estimate model parameters, our method uses a Monte Carlo EM algorithm to marginalize over the unknown ordering of mutations. We show that our method performs better on simulated data compared to current methods and leads to more parsimonious models. The application of proportional hazards to mutation processes is, to our knowledge, novel and formalizes the current methods in a statistical framework that can be easily extended to analyze the effect of other biological features on mutation rates.

1. Introduction. We introduce a proportional hazards model approach to study DNA mutation processes. Our study is motivated by somatic hypermutation, a mutation process that occurs in DNA sequences that encode B-cell receptors (BCRs), proteins that recognize and neutralize pathogens. When BCRs are secreted from B cells they are known as antibodies. The immune system relies on this somatic hypermutation process to generate a diversity of BCRs that can bind to

Received November 2017; revised September 2018.

[1]Supported by NIH Grants DP5OD019820 and T32CA206089.

[2]Supported by NIH Grant DP5OD019820.

[3]Supported by NIH Grants U19-AI117891 and R01-GM113246.

[4]Supported by R01-AI120961.

[5]Supported in part by a Faculty Scholar grant from the Howard Hughes Medical Institute and the Simons Foundation.

[6]Co-first authors.

[7]Cocorresponding authors.

Key words and phrases. Antibody maturation, survival analysis, Monte Carlo expectation–maximization, lasso, somatic hypermutation.

a large and continually evolving variety of pathogens. The starting material for this mutation process is a BCR sequence that is formed by recombination (Tonegawa (1983), Schatz and Ji (2011)). From this sequence a complex system of enzymes introduces mutations in a random pattern that is known to be highly sensitive to the sequence *motif*—the sequence of DNA bases surrounding the mutating position (Dunn-Walters et al. (1998), Chahwan et al. (2012), Rogozin and Kolchanov (1992), Methot and Di Noia (2017)).

Our goal is to develop a solid statistical framework that estimates the mutation rates of motifs and provides interpretable results for this mutation process. A better understanding of somatic hypermutation will help in designing vaccines for challenging viruses (Haynes et al. (2012), Hwang et al. (2017), Wiehe et al. (2018)), in furthering understanding of the biological mechanisms at play (Pham et al. (2003), Rogozin et al. (2001)) and in gaining insight into the natural selection process occurring in the immune system (Hershberg et al. (2008), Uduman et al. (2011), McCoy et al. (2015), Hoehn, Lunter and Pybus (2017)).

Several strategies have been used to model a motif's mutability, that is, how likely a position is to mutate given the motif at that position. The general approach is to compare a mutated sequence with its inferred ancestor sequence and model the differences between them. Cohen, Kleinstein and Louzoun (2011) and Elhanati et al. (2015) model the mutabilities of motifs as the product of the mutabilities of short subsequences (usually one or two bases). By using a log-linear model with only first-order terms, they keep the parameter count low but miss interactions between the positions. Yaari et al. (2013) and Cui et al. (2016) do not use this log-linear assumption; they allow a separate parameter for each possible five-nucleotide motif (of which there are 4^5) and use ad hoc methods to handle motifs with few observations. Rather than these restrictive and ad hoc approaches, a more data-adaptive variable selection method is desirable.

Another drawback of these methods is that they ignore mutations that occur in neighboring positions, even though such events can carry important information about highly mutable motifs. Indeed, these methods require counting the number of times a motif is observed to mutate. If mutations occur in neighboring positions, they cannot attribute the mutation to the correct motif. For settings with high rates of mutation, these methods end up estimating the mutabilities poorly. To properly estimate mutabilities, one needs to account for the different possible orders in which mutations occurred. Previous work has developed methods for performing various types of inference when this mutation order is unknown (Hwang and Green (2004), Hobolth (2008)), but these inference procedures make the parametric assumption that the mutation process follows a continuous time Markov process. Here, we relax this model assumption and use a semiparametric model instead.

In this paper we advance the modeling of motif mutabilities in several directions. We propose a method to fit mutabilities using survival analysis of mutation motifs called `samm`. We formalize the problem using Cox proportional hazards in which mutations are the failure events to be investigated. Although survival

models are used implicitly by computational immunologists for simulation (Yaari, Uduman and Kleinstein (2012), Sheng et al. (2017)), we believe this is the first time they have been used for inference.

To estimate motif mutabilities, our method uses the Monte Carlo expectation–maximization algorithm (MCEM, Wei and Tanner (1990)). Since the orders in which mutations occur are unobserved in our data, expectation–maximization (EM, Dempster, Laird and Rubin (1977)) allows us to perform maximum likelihood while averaging over these unknown orders. However, the E-step in EM requires calculating the expected log-likelihood which is analytically intractable since we must average over all possible mutation orders; thus, we estimate this expectation using Gibbs sampling. This approach is similar to that used by Goggins et al. (1998) to model interval-censored failure-time data where the order in which the failure events occur is unknown.

Our method also handles high-dimensional settings in which there are many more predictors than observations, which is important because many motifs are hypothesized to affect the mutation rate, but the specific ones are unknown. For instance, Yaari et al. (2013) and Cui et al. (2016) consider all motifs of length 5. We use the lasso (Tibshirani (1996)) to improve estimation and perform variable selection. To provide a measure of uncertainty of our estimates, we use a two-step approach. We fit an ℓ_1-penalized Cox proportional hazards model (Tibshirani et al. (1997)) to perform variable selection and refit an unpenalized model over the selected variables to obtain our final estimates along with approximate confidence intervals.

Section 2 describes our estimation methods, starting with a simplified logistic regression model and then progressing to our full estimation method. Section 3 presents simulation results. In Section 4 we apply our method to model somatic hypermutation of BCR sequences from Cui et al. (2016) and compare results with previous methods.

2. Methods. Our data consists of BCR nucleotide sequences that have mutated for an unknown period of time. Specifically, we target sequences that are undergoing mutation but not natural selection. Such data can be obtained, for example, through immunization experiments in transgenic mice designed to have a DNA segment that is carried along and mutated but not expressed as part of the BCR (Yeap et al. (2015), Cui et al. (2016)).

Though we focus on modeling the somatic hypermutation process of BCRs, our approach can be framed more generally as a problem of modeling a sequence-valued mutation process. We refer to the original, unmutated sequences as "naïve" and their descendants as "mutated" sequences. Throughout, we suppose that these naïve sequences are known. In the BCR case we restrict our attention to a computationally-identified naïve segment coded in germline DNA (the V region, Yaari and Kleinstein (2015)).

More formally, the mutation process of a sequence with p positions can be described as a vector-valued stochastic process $\{X(t) = (X_1(t), \ldots, X_p(t)) : t \in [0, \infty)\}$ indexed by time t. Each $\{X_j(t)\}$ represents the mutation process of the jth position in the sequence. For a given time t the state space of $X_j(t)$ is the set of nucleotides $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, and the state space of $X(t)$ is the set of length-p nucleotide sequences $\mathcal{S} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^p$. At the start of the mutation process, $X(0)$ is fixed to be the naïve sequence.

In a context-sensitive model the probability that a position mutates at time t depends on the current nucleotide sequence $X(t)$. In our work we assume that only local context matters: The mutation rate at each position is affected only by the local nucleotide sequence called the motif. For motif m we denote the length of the motif as $\text{len}(m)$, where $\text{len}(m)$ is typically much smaller than the number of nucleotides in $X(t)$. The function $I(X(t), m, j, j')$ is the binary indicator of whether motif m appears in sequence $X(t)$ from positions $j - j' + 1$ to $j - j' + \text{len}(m)$. More formally, it is defined as

$$I\big(X(t), m, j, j'\big) = \prod_{k=1}^{\text{len}(m)} 1\big\{X_{j-j'+k}(t) = m_k\big\}, \tag{1}$$

where m_k is the nucleotide in the kth position of motif m. This is known as a $\text{len}(m)$-mer, that is, a motif of length $\text{len}(m)$. For example, a 5-mer is a motif of length 5. In the special case where $\text{len}(m)$ is odd and $j' = (\text{len}(m) + 1)/2$, (1) checks if $X(t)$ has motif m centered at position j. We call this a centered motif; for all other cases we say that (1) is checking for an offset motif.

Define a *motif dictionary* to be a set $\mathcal{M}$ of sequence features (m, j') that may affect mutation rate. Example dictionaries include 1-mers (all length 1 motifs), offset 2-mers (length 2 motifs with $j' = 1, 2$), all of the central and offset 3-mers (length 3 motifs, with $j' = 1, 2, 3$) and all of the central 5-mers. We may also consider all possible unions of these dictionaries. Suppose we have selected a set $\mathcal{M}$. To ease exposition, we choose an arbitrarily assigned but fixed order $\{(m^{(1)}, j'^{(1)}), \ldots, (m^{(q)}, j'^{(q)})\}$ where q is the number of motif features in the dictionary $\mathcal{M}$. We may now define a function that indicates which elements in $\mathcal{M}$ occur at each position. For each position j let $\psi_j : \mathcal{S} \mapsto \{0, 1\}^q$ be defined by $[\psi_j(X(t))]_k \equiv I(X(t), m^{(k)}, j, j'^{(k)})$ for $k = 1, \ldots, q$. We use ψ_j as the feature vector for modeling the mutation rate of position j (Figure 1).

Of course the framework we present here generalizes to other types of dictionaries, including dictionaries that only specify bases for a subset of positions, but we will restrict to the above-described dictionaries in this paper for concreteness.

2.1. *Logistic regression.* As a simplified approach to modeling the mutation process, one may ignore the time component and use logistic regression. In this model each position in the sequence is independent and the probability of each

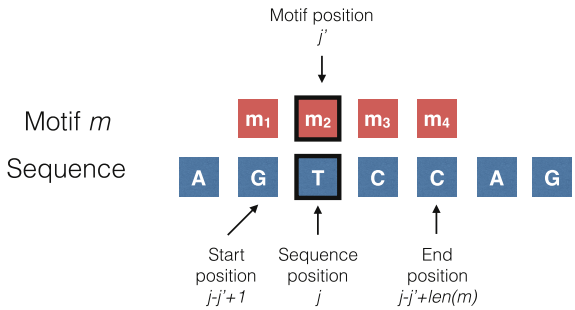


FIG. 1. *An example of how feature vectors are generated*: *if we believe that the mutation rate at a position depends on the* 4*-mer* (*i.e.*, *length* 4 *motif*) *starting one position to its left*, *then the feature vector for position* j *is a one-hot encoding of the sequence that appears in position* $j-1$ *through* $j+2$. *More formally*, *each element in the feature vector at position* j *indicates whether or not a motif* m *appears from start position* $j-j'+1$ *through end position* $j-j'+\text{len}(m)$ (*here* $m=4$ *and* $j'=2$). *The start and end positions are derived by aligning position* j *of the sequence with position* j' *of the motif*.

position mutating only depends on the *initial* nucleotide sequence $X(0)$, that is,

$$\text{(2)}\qquad \Pr(\text{mutation at position } j) = \frac{1}{1+\exp(-\boldsymbol{\theta}^\top \psi_j(X(0)))} \qquad \forall j \in \{1,\dots,p\}.$$

Yaari et al. (2013) essentially take this approach; the logistic model here just formalizes their intuition within a statistical framework and allows us to generalize their method to be applicable for any feature vector mapping. Moreover, we can use penalized logistic regression for handling high-dimensional models and encode various structural assumptions regarding the mutation rates; we discuss this in detail later in Section 2.4.

Logistic regression ignores the time component in a mutation process and as such ignores how the mutation rate of each position may change as other positions mutate (Figure 2). The assumption that the mutation rate only depends on the initial nucleotide sequence is most problematic when the mutation rate is high. Also, logistic regression ignores censoring; the method estimates the average mutation probability with respect to a particular sampling process. The estimates will be different if we tend to sample sequences that mutate for long vs. short periods of time. The following section addresses these issues by modeling the mutation process using a survival analysis framework.

2.2. *Cox proportional hazards*. We propose using a survival analysis framework to model the mutation process. We view positions in a single sequence as subjects observed for the same time period. A mutation event at position j occurs at time t if the nucleotide immediately before time t, $\lim_{s\to t^-} X_j(s)$, differs from the nucleotide at time t, $X_j(t)$. If a position never mutates, we consider its

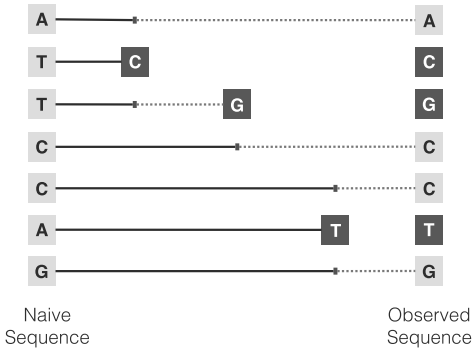


FIG. 2. *Survival analysis for BCR sequences where the positions that have not mutated are indicated by light gray squares and those that have mutated are indicated by dark gray squares. In a context-dependent mutation model a mutation event can change the mutation rates of other positions. Suppose the hazard (i.e., mutation) rate of a position depends on the position's two neighboring bases. Then, for example, when the* T *in the third position mutates to a* G*, the hazard rate for* C *in the fourth position changes from the original* TCC *motif to the* GCC *motif. Changes in the motif at a potential mutating position, and thus its hazard rate, are indicated by a change from solid to dashed lines.*

mutation time to be censored.[8] The hazard (or mutation) rate of a position is the instantaneous risk of mutating at time t given that it has been conserved up to time t. In between successive mutation times each position has a constant hazard rate and mutates independently from all other positions. The dependence between positions is introduced when a mutation occurs; upon a mutation event, the hazard rate for each neighboring position can change (Figure 2).

Accounting for how the sequence can change over time complicates our estimation procedure. Since we do not observe the order of mutation events in the data—we only observe pairs of naïve and mutated sequences—there are many possible mutation orders that could explain how the mutated sequence arose from the naïve sequence; each mutation order corresponds to a distinct sequence of hazard rates.

For ease of exposition we present our estimation method for the mutation process of a single pair of naïve and mutated nucleotide sequences. The method readily applies to estimating rates given many independent mutation processes (a typical application will be to thousands or more sequences).

As part of our modeling framework, we assume that each position can mutate at most once during the mutation process. This is a simplification of the somatic

[8]By using a survival analysis framework we implicitly assume that a mutation will occur at every position given a sufficiently long period of time. This assumption is reasonable for somatic hypermutation; the complex system of enzymes has the ability to mutate any position along the sequence (Chahwan et al. (2012)). This assumption may not hold for other DNA mutation processes, and the method may need to be modified accordingly.

hypermutation process since it is possible for a position to mutate more than once, though in our data the naïve and mutated sequence typically differ in 1–5% of the positions. We think this assumption is reasonable and makes the problem easier to handle from a computational standpoint. We discuss how this assumption affects performance under model misspecification in Section C.2 of the Supplementary Material (Feng et al. (2019)).

We model the hazard rate of position j using Cox proportional hazards, which supposes that the hazard rate j at time t is assumed to be of the form

$$h_j(t) = h_0(t)\exp\bigl(\boldsymbol{\theta}^\top \psi_j\bigl(X(t)\bigr)\bigr), \tag{3}$$

where $\boldsymbol{\theta} \in \mathbb{R}^q$ and the baseline hazard rate $h_0(\cdot)$ is an arbitrary unspecified baseline hazard function. Extending (3), we can additionally model the rate at which our process mutates to a specific nucleotide—the *target nucleotide*. Previous work (Cowell and Kepler (2000), Rogozin et al. (2001), Yaari et al. (2013), Cui et al. (2016)) suggests that the context-dependent mutation process is biased in favor of mutations to particular bases. We can take into account these preferences by considering a *per-target model*. In such a model we additionally define vectors $\boldsymbol{\theta}_{\mathtt{N}}$ for each possible target nucleotide $\mathtt{N} \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. Using a competing events framework, the rate of mutating to nucleotide $\mathtt{N}$ at position j at time t is modeled as

$$h_{j,\to \mathtt{N}}(t) = 1\bigl\{X_j(t) \neq \mathtt{N}\bigr\} h_0(t)\exp\bigl((\boldsymbol{\theta} + \boldsymbol{\theta}_{\mathtt{N}})^\top \psi_j\bigl(X(t)\bigr)\bigr). \tag{4}$$

As $\mathtt{N} \to \mathtt{N}$ is not considered a mutation, we include the indicator function $1\{\cdot\}$ in (4) to specify that a position cannot mutate to the nucleotide that currently appears there.

2.3. *Maximum likelihood via MCEM.* We are now ready to present a maximum likelihood estimation method for our model. We assume that the hazard rate follows (3). The per-target model in (4) is a straightforward extension of this simpler case. Let the observed data, namely the single pair of naïve and mutated nucleotide sequences, be denoted $\mathbf{S}_{\text{obs}}$, where we suppose that n positions have mutated.

When $h_0(t)$ is an arbitrary unspecified baseline hazard function, only the order of the mutations carries information about $\boldsymbol{\theta}$, even if the mutation times are observed (Kalbfleisch and Prentice (2011)). Explained intuitively, time can be transformed by an arbitrary increasing function, and the form of the hazard function would still be of the form (3) (for more details, see Chapter 4 in Kalbfleisch and Prentice (2011)). Consequently, estimating $\boldsymbol{\theta}$ involves only maximizing the likelihood of observing the mutation order.

For now, suppose we observe the order in which the mutations occurred. Let π_j be the position of the jth mutation for $j = 1, \ldots, n$. Let $\boldsymbol{\pi}_{1:j}$ denote the positions of the first through jth mutation, where $\boldsymbol{\pi}_{1:0}$ is defined to be the empty set. Define

$S(\boldsymbol{\pi}_{1:j})$ to be the nucleotide sequence after positions $\boldsymbol{\pi}_{1:j}$ mutate. Thus, the observed data is $\mathbf{S}_{\text{obs}} = \{S(\pi_{1:0}), S(\pi_{1:n})\}$. The set $R(\boldsymbol{\pi}_{1:j}) \equiv \{1, \ldots, p\} \setminus \boldsymbol{\pi}_{1:j}$ is the set of positions at risk of mutating, commonly referred to as the risk group in the survival analysis literature. Then the marginal likelihood of $\boldsymbol{\theta}$ is

$$\mathcal{L}_c(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}; \boldsymbol{\theta}) = \prod_{j=1}^{n} \frac{\exp(\boldsymbol{\theta}^\top \psi_{\pi_j}(S(\boldsymbol{\pi}_{1:j-1})))}{\sum_{k \in R(\boldsymbol{\pi}_{1:j-1})} \exp(\boldsymbol{\theta}^\top \psi_k(S(\boldsymbol{\pi}_{1:j-1})))}. \tag{5}$$

Our result looks like the marginal likelihood derived in equation 4.47 in (Kalbfleisch and Prentice (2011)) except that it is derived under a more general set of assumptions; whereas they assume the covariates are fixed, we assume the covariates to be fixed between events. The derivation of (5) is given in the Supplementary Material (Feng et al. (2019)).

The marginal likelihood in (5) implies that the mutation order can be simulated by drawing positions from successive multinomial distributions. To simulate mutation at the jth position, we draw a position from the risk group $R(\boldsymbol{\pi}_{1:j-1})$. In fact, Gupta et al. (2015) use this procedure to simulate the somatic hypermutation process, though they do not provide a statistical justification.

Unfortunately, the mutation order $\boldsymbol{\pi}$ is not observed in our problem. We instead maximize the observed data likelihood which is the complete data likelihood marginalized over all admissible mutation orders $\mathcal{A}(\mathbf{S}_{\text{obs}})$:

$$\mathcal{L}(\mathbf{S}_{\text{obs}}; \boldsymbol{\theta}) = \sum_{\boldsymbol{\pi} \in \mathcal{A}(\mathbf{S}_{\text{obs}})} \mathcal{L}_c(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}; \boldsymbol{\theta}). \tag{6}$$

Assuming positions mutate at most once, $\mathcal{A}(\mathbf{S}_{\text{obs}})$ is a set of $n!$ possible mutation orders. When the number of mutated positions n is small, we can enumerate all possible mutation orders and maximize (6) using a nonlinear optimization algorithm such as EM (Dempster, Laird and Rubin (1977)). However, in most data sets n is much too large for direct enumeration to be computationally tractable, so we maximize (6) using MCEM.

MCEM extends the traditional EM algorithm by approximating the expectation in the E-step using a Monte Carlo sampling method. Let $\boldsymbol{\pi} = \boldsymbol{\pi}_{1:n}$ be a full mutation order. We use the Gibbs sampler in Algorithm 1 to sample $\boldsymbol{\pi} \mid \{\mathbf{S}_{\text{obs}}, \boldsymbol{\theta}\}$. Given a full mutation order $\boldsymbol{\pi}$, let $\boldsymbol{\pi}_{(-j)}$ be the partial mutation order where the jth mutation is removed from $\boldsymbol{\pi}$; a full mutation order $\boldsymbol{\pi}'$ is consistent with $\boldsymbol{\pi}_{(-j)}$ if there is some $j' \in \{1, \ldots, n\}$ such that $\boldsymbol{\pi}'_{(-j')} = \boldsymbol{\pi}_{(-j)}$. For instance if $\boldsymbol{\pi} = [1, 3, 2]$, then the partial mutation order $\boldsymbol{\pi}_{(-2)}$ is $[1, 2]$, and $\boldsymbol{\pi}' = [3, 1, 2]$ is consistent with $\boldsymbol{\pi}_{(-2)}$ since $\boldsymbol{\pi}_{(-2)} = \boldsymbol{\pi}'_{(-1)}$. For each Gibbs sweep the index j cycles through $\{1, \ldots, n\}$ in some random order. For Gibbs step k we sample a full mutation order $\boldsymbol{\pi}^{(k)}$ that is consistent with the partial mutation order $\boldsymbol{\pi}^{(k-1)}_{(-j)}$. The proof that this sampler converges to the desired probability distribution is standard and similar to that of Goggins et al. (1998).

We efficiently calculate the probability of a full mutation order given a partial mutation order by reusing previous computations. In particular, for partial mutation order $\boldsymbol{\pi}_{(-j)}$, we calculate the probabilities of each consistent full mutation order starting from the full mutation order where position π_j mutates first to that where position π_j mutates last. By ordering consistent full mutation orders in this way, the j'th consistent full mutation order $\boldsymbol{\pi}'$ and $(j'+1)$th consistent full mutation order $\boldsymbol{\pi}''$ are the same except that the j' and $(j'+1)$th mutations are swapped. The ratio of the conditional probabilities of $\boldsymbol{\pi}'$ and $\boldsymbol{\pi}''$ given $\boldsymbol{\pi}_{(-j)}$ is

$$\begin{aligned}\frac{\Pr(\boldsymbol{\pi}'|\boldsymbol{\pi}_{(-j)})}{\Pr(\boldsymbol{\pi}''|\boldsymbol{\pi}_{(-j)})} &= \frac{\exp(\boldsymbol{\theta}^\top(\psi_{\pi'_{j'}}(S(\boldsymbol{\pi}'_{1:j'-1})) + \psi_{\pi'_{j'+1}}(S(\boldsymbol{\pi}'_{1:j'}))))}{\exp(\boldsymbol{\theta}^\top(\psi_{\pi''_{j'}}(S(\boldsymbol{\pi}''_{1:j'-1})) + \psi_{\pi''_{j'+1}}(S(\boldsymbol{\pi}''_{1:j'}))))} \\ &\quad \times \frac{\sum_{i\in R(\boldsymbol{\pi}''_{1:j'})}\exp(\boldsymbol{\theta}^\top\psi_i(S(\boldsymbol{\pi}''_{1:j'})))}{\sum_{i\in R(\boldsymbol{\pi}'_{1:j'})}\exp(\boldsymbol{\theta}^\top\psi_i(S(\boldsymbol{\pi}'_{1:j'})))}. \qquad (7)\end{aligned}$$

So if we already have $\Pr(\boldsymbol{\pi}''|\boldsymbol{\pi}_{(-j)})$, we can divide it by (7) to quickly obtain $\Pr(\boldsymbol{\pi}'|\boldsymbol{\pi}_{(-j)})$. Moreover, we can efficiently calculate (7) by storing previous computational results; for instance, the summation over the risk group $R(\boldsymbol{\pi}'_{1:j'})$ shares many elements with the summation over the risk group $R(\boldsymbol{\pi}''_{1:j'})$. Similar ideas can be used to speed up other calculations required for MCEM.

Algorithm 1 Gibbs sampler for mutation orders

```
Initialize Gibbs step index k = 1 and mutation order π^(0).
for Gibbs sweep index i = 1, 2, ... do
  for j ∈ {1, ..., n} do
    π_(−j) := π^(k−1)_(−j)
    Sample π^(k) from the distribution Pr(π | π^(k−1)_(−j)).
    k := k + 1
  end for
end for
```

Given the Monte Carlo samples from the E-step, the M-step maximizes the mean log-likelihood of the complete data. Suppose the E-step generates Monte Carlo samples $\boldsymbol{\pi}^{(1)}, \ldots, \boldsymbol{\pi}^{(E)}$. Then, during the M-step we solve

$$\max_{\boldsymbol{\theta}} \frac{1}{E}\sum_{i=1}^{E} \log \mathcal{L}_c(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}^{(i)}; \boldsymbol{\theta}) \qquad (8)$$

using iterative procedures such as gradient ascent.

We use ascent-based MCEM (Caffo, Jank and Jones (2005)) to maintain the monotonicity property of the EM algorithm. Briefly, ascent-based MCEM gives

a rule for deciding if the proposed MCEM estimate at each iteration should be accepted or if the Monte Carlo sample size should be increased. As the number of Monte Carlo samples increases, the standard error of the estimated expected log-likelihood decreases. So for a sufficiently large number of Monte Carlo samples, we can ensure that the observed data likelihood increases with high probability.

2.4. *Regularization and variable selection*. In many cases it is desirable to model the effects of many features. For instance Yaari et al. (2013) estimate a 5-mer model with 1024 parameters. Estimating the parameters for a per-target model increases the number of parameters by an additional factor of four. If the number of sequences in the dataset is small compared to the number of features, the optimization problem in (6) can be ill posed. For such high-dimensional settings it is common to use regularization to stabilize our estimates and encourage model structure.

In particular, we may believe that only a small subset of the features affects the mutation rate. Yaari et al. (2013) assume that the nucleotides closest to a position have the most significant effect on its mutation rate. For 5-mer motifs with a small number of observations, they estimate its mutation rate using an offset 3-mer motif. In our method we use the lasso (Tibshirani (1996)) to perform variable selection.

To incorporate the lasso, our estimation procedure requires two steps. The first step maximizes the observed log-likelihood with a lasso penalty and thereby performs variable selection. The second step aims to quantify the uncertainty of our model parameter estimates. We refit the model parameters by maximizing the unpenalized objective and use the confidence intervals for the unpenalized model as an assessment of uncertainty.

In the first step, we split the data into training and validation sets denoted $\mathbf{S}_{\text{obs,train}}$ and $\mathbf{S}_{\text{obs,val}}$ respectively, and maximize the penalized log-likelihood of the training data

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \{\log \mathcal{L}(\mathbf{S}_{\text{obs,train}}; \boldsymbol{\theta}) - \lambda \|\boldsymbol{\theta}\|_1\}, \tag{9}$$

where $\lambda > 0$ is a penalty parameter. To solve (9), we use a variant of MCEM: the E-step is the same as before, but we maximize the penalized EM surrogate function during the M-step. The penalized EM surrogate function is simply (8) with a lasso penalty:

$$\frac{1}{E}\sum_{i=1}^{E} \log \mathcal{L}_c(\mathbf{S}_{\text{obs,train}}, \boldsymbol{\pi}^{(i)}; \boldsymbol{\theta}) - \lambda \|\boldsymbol{\theta}\|_1. \tag{10}$$

This can be maximized using the generalized gradient ascent algorithm given in Algorithm 2 (Beck and Teboulle (2009), Nesterov (2013)).

We tune the penalty parameter λ in (9) by training-validation split. In the typical ideal case, we choose the penalty parameter that maximizes the likelihood of the

Algorithm 2 M-step via generalized gradient ascent

Initialize $\boldsymbol{\theta}$. Choose a step size $\alpha > 0$.
for iteration $k = 1, 2, \ldots$ until convergence **do**

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha \nabla_\theta \frac{1}{E} \sum_{i=1}^{E} \mathcal{L}_c\big(\mathbf{S}_{\text{obs,train}}, \boldsymbol{\pi}^{(i)}; \boldsymbol{\theta}\big)$$

  for parameter index $j = 1, .., p$ **do**
    $\theta_j := \operatorname{sign}(\theta_j) \max(|\theta_j - \lambda|, 0)$
  end for
end for

observed validation data. Unfortunately, the likelihood of observed data is computationally intractable. Instead, we use the property that, for any $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, the difference between the log-likelihoods of the observed data is bounded below by the difference between the expected log-likelihoods of the complete data

$$\begin{aligned} &\log \mathcal{L}(\mathbf{S}_{\text{obs}}; \boldsymbol{\theta}) - \log \mathcal{L}\big(\mathbf{S}_{\text{obs}}; \boldsymbol{\theta}'\big) \\ &\quad \geq \mathbb{E}\big[\log \mathcal{L}_c(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}; \boldsymbol{\theta}) - \log \mathcal{L}_c\big(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}; \boldsymbol{\theta}'\big) \mid \mathbf{S}_{\text{obs}}; \boldsymbol{\theta}'\big], \end{aligned} \tag{11}$$

which follows directly from Jensen's inequality. The expectation above is taken with respect to the conditional distribution of the mutation orders $\boldsymbol{\pi}$ given the observed data $\mathbf{S}_{\text{obs}}$ and model parameter $\boldsymbol{\theta}'$. Thus the right-hand side can be estimated by sampling mutation orders from the Gibbs sampler in Algorithm 1. If the right-hand side of (11) is positive, then $\boldsymbol{\theta}$ has a higher log-likelihood than $\boldsymbol{\theta}'$ on the validation set. However, if the right-hand side is negative, we do not know how the two parameters compare.

Our proposal for tuning the penalty parameter, Algorithm 3, is based on (11). The algorithm searches across a one-dimensional grid of penalty parameters, from largest to smallest. For consecutive penalty parameters we estimate the right-hand side of (11) to determine if the smaller penalty parameter has a higher observed log-likelihood. We keep shrinking the penalty parameter until the estimate for the right-hand side of (11) is negative. Since the check based on (11) is conservative, we may end up choosing a penalty parameter that is slightly larger than desired. Nonetheless, our simulation results suggest that this procedure works well in practice.

Algorithm 3 can be easily extended to incorporate multiple training-validation splits such as in k-fold cross validation. We average the estimates of the right-hand side of (11) across the training-validation splits and stop shrinking the penalty parameter if the average is negative. After selecting a penalty parameter we obtain the final parameter support from the k-fold procedure by refitting the penalized model on the whole training set.

Algorithm 3 Tuning penalty parameters via training-validation split

Consider a grid of penalty parameters $\lambda_1 > \cdots > \lambda_K \geq 0$.
Initialize $\lambda_{\text{best}} := \lambda_1$. Fit λ_1 to get $\hat{\boldsymbol{\theta}}_{(1)}$.
for iteration $i = 2, \ldots, K$ **do**
  Solve (9) with $\lambda \equiv \lambda_i$ to get $\hat{\boldsymbol{\theta}}_{(i)}$.
  Estimate via Monte Carlo

$$\eta = \mathbb{E}\big[\log \mathcal{L}_c(\mathbf{S}_{\text{obs,val}}, \boldsymbol{\pi}; \hat{\boldsymbol{\theta}}_{(i)}) - \log \mathcal{L}_c(\mathbf{S}_{\text{obs,val}}, \boldsymbol{\pi}; \hat{\boldsymbol{\theta}}_{(i-1)}) \,|\, \mathbf{S}_{\text{obs,val}}; \hat{\boldsymbol{\theta}}_{(i-1)}\big]. \tag{12}$$

  if $\eta > 0$ **then**
    $\lambda_{\text{best}} := \lambda_i$
  else
    break
  end if
end for

Now, we move on to the second step where our goal is to quantify the uncertainty of our estimated model parameters. Unfortunately, estimating confidence intervals after model selection is a difficult problem, even in the much simpler case of linear models (Dezeure et al. (2015)). Hence, some papers use the approach of fitting a penalized model, refitting an unpenalized model based on the selected variables and then using the confidence intervals generated using traditional inference procedures for unpenalized models (Leeb, Pötscher and Ewald (2015), Hesterberg et al. (2008)). We proceed in the same manner: we refit the model by maximizing the unpenalized observed log-likelihood (6) of the entire dataset with respect to the selected variables and constraining the others to zero; then, we construct confidence intervals for the unpenalized model, ignoring the fact that we have already peeked at the data in the first step. Though these confidence intervals are asymptotically valid only under very restrictive conditions, they provide some measure of the uncertainty of our fitted parameters; we show via simulation in Section 3 that the coverage of these intervals is close to nominal. To highlight that these intervals are not truly confidence intervals, we refer to them as uncertainty intervals, where $100(1-\alpha)\%$ uncertainty intervals are constructed using intervals with nominal $100(1-\alpha)\%$ coverage.

To obtain these uncertainty intervals, we calculate the standard error of our estimates using an estimate of the observed information matrix. Louis (1982) shows that the observed information matrix is related to the complete data likelihood via the following identity:

$$I[\boldsymbol{\theta} \mid \mathbf{S}_{\text{obs}}] = -\mathbb{E}\big[\nabla^2_{\boldsymbol{\theta}} \log \mathcal{L}_c(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}; \boldsymbol{\theta}) \mid \mathbf{S}_{\text{obs}}; \boldsymbol{\theta}\big]$$

$$- \mathbb{E}\big[\nabla_{\boldsymbol{\theta}} \log \mathcal{L}_c(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}; \boldsymbol{\theta}) \big(\nabla_{\boldsymbol{\theta}} \log \mathcal{L}_c(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}; \boldsymbol{\theta})\big)^\top \mid \mathbf{S}_{\text{obs}}; \boldsymbol{\theta}\big]$$

$$+ \mathbb{E}\big[\nabla_{\boldsymbol{\theta}} \log \mathcal{L}_c(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}; \boldsymbol{\theta}) \mid \mathbf{S}_{\text{obs}}; \boldsymbol{\theta}\big] \mathbb{E}^\top\big[\nabla_{\boldsymbol{\theta}} \log \mathcal{L}_c(\mathbf{S}_{\text{obs}}, \boldsymbol{\pi}; \boldsymbol{\theta}) \mid \mathbf{S}_{\text{obs}}; \boldsymbol{\theta}\big].$$

Therefore, we can estimate the observed information matrix using samples from the final MCEM iteration and then invert it to obtain uncertainty intervals.

Finally, one caveat of our method is that the two-step procedure is not guaranteed to give estimates of standard errors/uncertainty intervals: The first step of our procedure may choose a penalty parameter such that the estimated information matrix in the second step is not positive definite. We see this behavior in a small number of simulations in Section 3, though we do not observe such behavior in our data analysis. To avoid this issue, we suggest combining k-fold cross-validation with Algorithm 3 and use the average estimate of the lower bound (12) from each of the k folds to tune the penalty parameter.

Our GPLv3-licensed Python implementation of `samm` is available at http://github.com/matsengrp/samm. The repository includes code used for generating plots in this manuscript, as well as a tutorial for how to run `samm`. All output from Sections 3 and 4 is available on http://zenodo.org/record/1321330 with DOI 10.5281/zenodo.1321330.

2.5. *Examples*. By varying the motif dictionary $\mathcal{M}$, our procedure can fit different models of the mutation process. In this section we list some example models that can be fit using our procedure and discuss the interplay between the motifs included in $\mathcal{M}$ and our feature-selection step. In the simplest case, analogous to existing work (Yaari et al. (2013), Cui et al. (2016)), we can estimate a "k-mer model" (where k is odd) by letting

$$\mathcal{M} = \big\{\big(m, (k+1)/2\big) : m \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^k\big\}. \tag{13}$$

The lasso would encourage setting elements in $\boldsymbol{\theta}$ to zero which means that these k-mer motifs would have the same baseline risk of experiencing a mutation.

In practice instead of modeling only the effects of k-mers for a fixed k, we may believe that the hazard rate for a position is affected more by positions closer to it. In this case we can model the effect of z-mers of varying length, for example, $1, 3, \ldots, k$-mer motifs, with

$$\mathcal{M} = \big\{\big(m, (z+1)/2\big) : m \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^z, z \in 1, 3, \ldots k\big\}. \tag{14}$$

We refer to this model as "hierarchical," as the elements in $\mathcal{M}$ relate to each other in a nested fashion. By including motifs in a hierarchical fashion, the lasso penalty encourages z-mers with the same inner $(z-2)$-mer to share the same mutation rate. This model formalizes the intuition used by Yaari et al. (2013). They try to estimate the mutation rates of 5-mers but fall back to using a 3-mer sub-motif if that 5-mer does not appear enough times in the data.

As mentioned before, we can add offset motifs to our motif dictionary as previous work suggests the mutation rates depend on upstream or downstream motifs

(Rogozin and Kolchanov (1992), Pham et al. (2003), Yaari et al. (2013)). For instance, one can include all the offset motifs that overlap the mutating position in the motif dictionary. We refer to such models as offset k-mer models.

Finally, we can model the hazard rate of motifs mutating to different target nucleotides as in (4). We parameterize the model using $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{\mathtt{N}}$ for $\mathtt{N} \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ since the penalized per-target model

$$\begin{aligned}&\arg \max_{\boldsymbol{\theta}, \boldsymbol{\theta}_{\mathtt{N}}: \mathtt{N} \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \log \mathcal{L}\big(\mathbf{S}_{\text{obs,train}}; \boldsymbol{\theta}, \{\boldsymbol{\theta}_{\mathtt{N}} : \mathtt{N} \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}\}\big) \\ &\quad - \lambda\Bigg(\|\boldsymbol{\theta}\|_1 + \sum_{\mathtt{N} \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \|\boldsymbol{\theta}_{\mathtt{N}}\|_1\Bigg) \qquad (15)\end{aligned}$$

will encourage hazard rates for the different target nucleotides to be the same if they share the same motif.

Many of these example models are over parameterized in order to obtain some desired sparsity pattern. Such over-parameterized models may have singular information matrices during the refitting procedure. However, this is not a problem since we are truly interested in the confidence intervals for the parameters $\boldsymbol{\theta}_{\text{agg}} = \boldsymbol{A}\boldsymbol{\theta}$ associated with the simple k-mer model, where $\boldsymbol{A}$ is a matrix that aggregates hierarchical motifs into a single k-mer. Since this aggregate k-mer model is identifiable, we can get uncertainty intervals for $\boldsymbol{\theta}_{\text{agg}}$. We calculate the pseudo-inverse $\boldsymbol{I}^-$ of the (estimated) information matrix and then use $\boldsymbol{A}\boldsymbol{I}^-\boldsymbol{A}^\top$ to get an estimate of the covariance matrix of $\boldsymbol{\theta}_{\text{agg}}$.

3. Simulation results. We now present a simulation study of our procedure, including a comparison to the current state-of-the-art method `SHazaM` (Yaari et al. (2013), version 0.1.8) and the logistic regression approach in Section 2.1.

3.1. *Understanding the effect of various models and settings.* We fit the following three models to simulated data:

- 3-mer model—the hazard rate modeled by (3) with motif dictionary (13) where $k = 3$,
- 3-mer per-target model—the hazard rate modeled by (4) with motif dictionary (13) where $k = 3$,
- 2,3-mer model—the hazard rate modeled by (3) with motif dictionary

$$\mathcal{M} = \big\{(m, j') : m \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^2, j' \in \{1, 2\}\big\} \cup \big\{(m, 2) : m \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^3\big\}.$$

To understand how dataset composition affects the performance of our procedure, we simulate different datasets by varying the sample sizes, sparsity levels and effect sizes.

We generate the true $\boldsymbol{\theta}^*$ according to the same hierarchical structure as each model we consider. Let the model parameters corresponding to the motif m be θ^*_m and corresponding to motif m with target nucleotide $\mathtt{N}$ be $\theta^*_{m\to\mathtt{N}}$. To obtain the

desired sparsity level, we randomly select a portion of the parameters to zero out. For per-target parameters, instead of setting the probability of mutating to N to zero, we set $\theta^*_{m\to \mathtt{N}}$ to $\log 1/3$ for all possible values of N, indicating no mutation preference. We scale the model parameters appropriately to control the effect size.

Our goal with these simulations is to obtain synthetic data that reflects different possible settings one may encounter when analyzing experimental data. We use the experimental data in Cui et al. (2016) analyzed in Section 4 as a template and alter various underlying properties of this dataset to simulate data that replicates what typical real-world datasets look like. We first generate naïve sequences using `partis`[9] (Ralph and Matsen IV (2016a, 2016b)) by drawing a set of genes from the IMGT database (Lefranc et al. (1999)) and simulating an observation frequency for each. Antibodies are composed of two units, a heavy and a light chain. Further, light chains can be classed as either κ or λ depending on where the encoded sequence came from in the genome. Both mice and humans have antibodies structured in this way. We select only κ-light chain mouse BCRs for our simulation, as this reflects our experimental data in Section 4. To generate the true $\boldsymbol{\theta}^*$ parameters, we randomly draw values from the mouse somatic hypermutation targeting model `MK_RS5NF` of Cui et al. (2016); we refer to these parameters as $\boldsymbol{\theta}^*_{\mathtt{MK}}$. The `MK_RS5NF` model is a collection of mutabilities and substitution probabilities from a 5-mer fit to κ-light chain mouse BCR data.

The average length of the naïve sequences is around 290 nucleotides. We use the survival model to mutate between 1% and 5% of the positions of each naïve sequence, obtaining a collection of simulated BCR sequences. Conditional on their naïve sequences, BCR sequences mutate independently.

We vary sparsity, effect size and sample size as follows. We generate the true $\boldsymbol{\theta}^*$ parameters with 25%, 50% and 100% nonzero elements. We also consider different effect sizes by scaling $\boldsymbol{\theta}^*$ such that its variance is 50%, 100% and 200% of the variance of the values in $\boldsymbol{\theta}^*_{\mathtt{MK}}$. Finally, we fit the model using 100, 200 and 400 mutated BCR sequences. For the main manuscript we report the simulation settings where we vary one simulation setting and fix the other settings to the middle value (e.g., we vary number of samples but keep the effect size at 100% and the number of nonzero elements at 50%); we report the result from running 100 replicates for each setting. For the remaining possible settings, as each separate model fit takes on average an hour to complete, we run only 10 replicates and report the results in Section C.3 of the Supplementary Material (Feng et al. (2019)).

To determine the optimal penalty parameter for `samm`, we split the data by gene subgroups, an externally-defined categorization that groups genes that share at least 75% identity at the nucleotide level (Lefranc (2014)), reserving 20% of subgroups for validation and the remainder for training. Splitting by gene subgroup ensures that the training and validation sets look sufficiently different; otherwise, the

[9]Version 0.12.0: http://git.io/fNvOx.

sequences in the validation set look nearly identical to those in the training set, and we select a penalty parameter that is too small. We then apply Algorithm 3 over a decreasing sequence of penalty parameter values $10^{-j}, 10^{-(j+0.5)}, 10^{-(j+1)}, \ldots$. The starting value for the sequence of penalty parameter values was pretuned so that we use a smaller j for smaller effect sizes and sample sizes. In particular, we chose $j = 1$ if effect size is 50% or sample size is 100; $j = 2$ if the effect size is 200% or sample size is 400, and $j = 1.5$ otherwise.

For each penalty parameter value we run a maximum of 10 MCEM iterations. Mutation orders are sampled from each Gibbs sampler run every eight sweeps after an initial burn-in period of 16 Gibbs sweeps. For each E-step we sample four mutation orders and continue to double the number of sampled mutation orders if the proposed estimate is not accepted by ascent-based MCEM. Once we have an estimate of the support of our model, we refit an unpenalized model to obtain uncertainty estimates. We run MCEM until the model has converged, and the variance estimates of the estimated model parameters are all nonnegative.

We assess the performance of our procedure using three measures. These performance metrics are all calculated with respect to the aggregate model since our complete model is overparameterized by design. We calculate the relative $\boldsymbol{\theta}$ error, defined as $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 / \|\boldsymbol{\theta}^*\|_2$, to see how close the estimated parameters are to the true parameter $\boldsymbol{\theta}^*$. We also calculate Kendall's tau coefficient to see how well our procedure ranks the motifs in terms of their mutabilities. Finally, we calculate the coverage of our approximate 95% uncertainty intervals. We define the average coverage as the proportion of aggregate model parameters where the uncertainty intervals covered the true value. The coverage calculations only involve aggregate parameters not zeroed out by our models.

These simulations demonstrate that our estimation procedure performs as expected (Figure 3). As the sample size and effect size increase the relative $\boldsymbol{\theta}$ error decreases, and the rank correlation increases. On the other hand as the percent of nonzero elements increases, both the relative $\boldsymbol{\theta}$ error and rank correlation increase. The error increases because there are more parameters to estimate. The increase in rank correlation is likely an artifact of how the metric is calculated, as Kendall's tau removes ties from the calculations. In particular, as the percent of nonzero elements increases, the number of ties in the data decreases, so the rank correlation seems to increase. In all the plots we see that the 3-mer per-target model tends to be the most difficult to estimate. This is expected as it contains 256 parameters whereas the 3-mer model only has 64 parameters.

Our simulations show that the coverages for the 3-mer and the 2,3-mer models are close to 95%, which is surprising as our uncertainty intervals ignore the double-peeking issue (Figure 3). Zhao, Shojaie and Witten (2017) explain why this procedure might work. Under certain assumptions the variables selected by

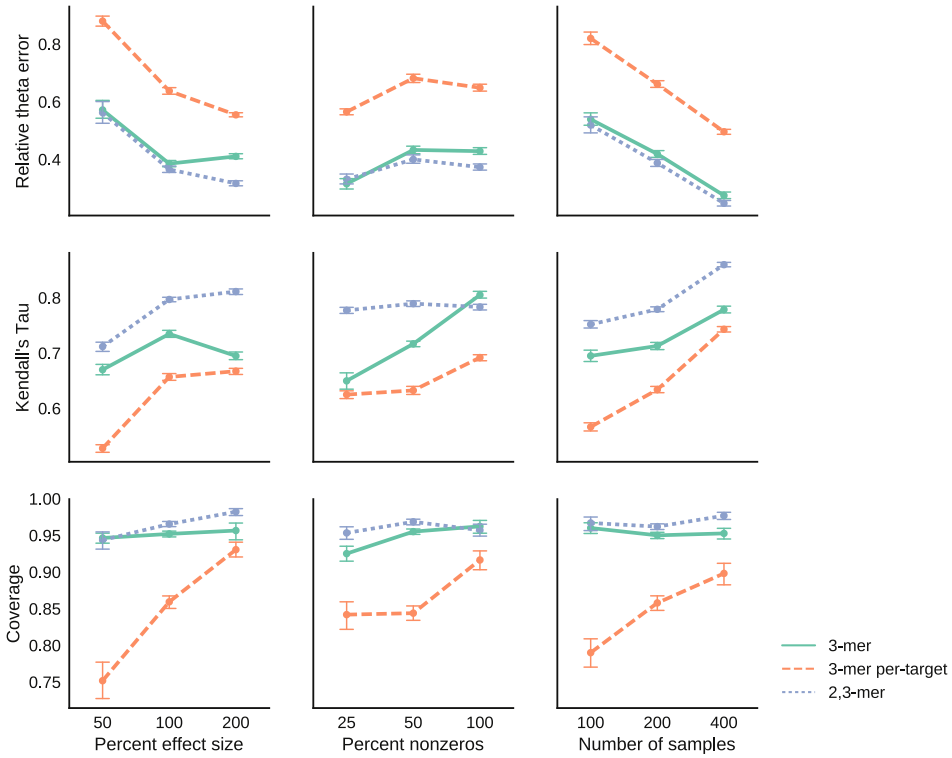


FIG. 3. *Relative error, correlation and coverage under different simulations settings for* 3*-mer,* 3*-mer per target and* 2,3*-mer models.*

the lasso are deterministic with high probability, so using the lasso to select variables does not really constitute as peeking at the data twice.

However, the coverage of the 3-mer per-target is much lower, dropping below 80% in certain settings (Figure 3). We suspect that the low coverage is mainly due to a lack of data, as the coverage improves with the number of samples. When there is a small number of samples compared to the number of parameters, our method may only provide a reasonable ranking of how mutable the motifs are but may not provide good estimates and uncertainty intervals.

Across the 2700 simulation runs there were 20 where the estimated information matrices were not positive definite and, therefore, uncertainty intervals cannot be calculated (Table 3). We believe that this occurs when the selected penalty parameter is too small; for small penalty parameters, the support of the fitted model becomes too large. In this case, when we refit the model with no penalty parameter, the problem is ill posed, and therefore, the estimated information matrix is not positive definite. To avoid this issue, we recommend using k-fold cross-validation in practice rather than just a training/validation split. (We use 5-fold cross-validation for the real data analysis and do not run into this issue.)

3.2. *Method comparisons*. In this section we compare the performance of `samm` to `SHazaM` and penalized logistic regression on simulated data. Since `SHazaM` only estimates the effect of 5-mer motifs, we simulate data such that the mutation rate at a specific site depends on the 5-mer centered at that position and the target nucleotide. We simulate 2000 BCR sequences from four mice. For each mouse we generate a separate set of naïve sequences using the same procedure as in Section 3.1. From these naïve sequences we simulate the mutation process independently to generate BCR sequences. We use two methods to simulate the mutation process:

- *Survival Simulation*: We generate model parameters $\boldsymbol{\theta}$ by resampling the values from $\boldsymbol{\theta}_{\mathtt{MK}}$ into a 3,5-mer per-target model structure. We then mutate the naïve sequences according to the survival model.
- *`SHMulate` Simulation*: We use $\boldsymbol{\theta}_{\mathtt{MK}}$ and mutate the naïve sequences using the `SHMulate` function in the `SHazaM` package (Yaari et al. (2013), Gupta et al. (2015)). `SHMulate` simulates the mutation process using a procedure that is similar to a survival model. However the exact calculations differ somewhat (e.g., it does not allow the mutation process to create stop codons).

`SHazaM` should have an advantage in the `SHMulate` simulations since the $\boldsymbol{\theta}_{\mathtt{MK}}$ was estimated using `SHazaM` on a separate BCR dataset and `SHazaM` uses some prior assumptions about the model structure. In particular, `SHazaM` assumes that 5-mer motifs that share certain upstream/downstream nucleotides have similar mutabilities. The simulations are run until 1–5% of the sequence is mutated. This mutation rate is on the low end for affinity-matured BCR sequences (compare the $3\times$ higher rate in He et al. (2014)), giving `SHazaM` and logistic regression a slight edge since the mutation rates will not change for most positions with accumulation of BCR mutations.

We fit a 3,5-mer per-target `samm` model using the same procedure as in Section 3.1. Using the same motif dictionary, we also fit a 3,5-mer per-target logistic regression model using logistic regression with a lasso penalty. We measure model performance by the relative $\boldsymbol{\theta}$ error and rank correlation over 100 simulation replicates.

Our method implemented in `samm` significantly outperforms logistic regression and `SHazaM` in both scenarios (Table 1), even though `SHazaM` should have an advantage when we simulate data using a dense model from `SHMulate`. Logistic regression and `SHazaM` tended to produce similar estimates, though logistic regression tended to do better when we simulated using the survival model and `SHazaM` tended to do better when we used the `SHMulate` model.

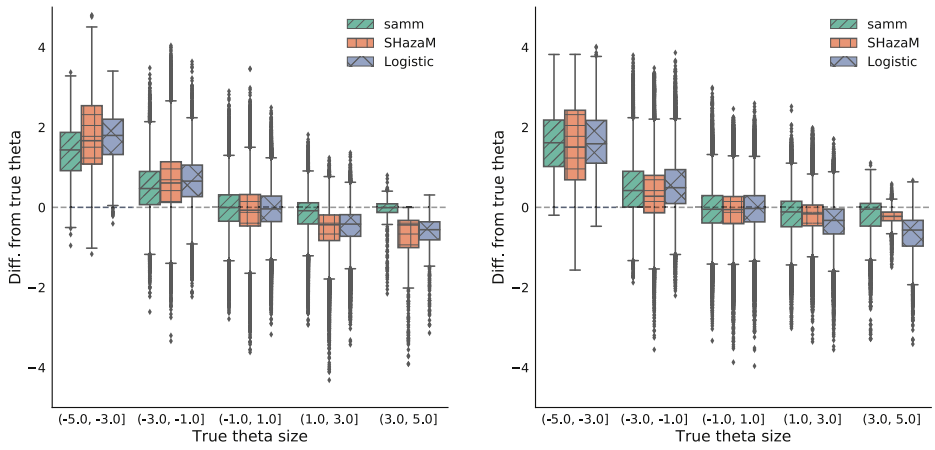
We present the results of model fitting in more detail in Figure 4. For negative $\boldsymbol{\theta}$ values all the methods are biased toward zero, though `SHazaM` and logistic regression tend to be more so. For positive $\boldsymbol{\theta}$ values `samm` is nearly unbiased while `SHazaM` and logistic regression are somewhat biased toward zero. The methods probably have trouble estimating negative values since we only observe a small

TABLE 1
Comparison of `samm`, `SHazaM`, *and penalized logistic regression given* 2000 *simulated B-cell receptor sequences from four mice. Relative* $\boldsymbol{\theta}$ *error and Kendall's tau computed separately for each of the* 100 *replicates. Monte Carlo standard errors calculated over these* 100 *estimates are given in parentheses*

Simulator	Model	Relative $\boldsymbol{\theta}$ error	Kendall's tau
survival model	`samm`	0.571 (0.002)	0.630 (0.001)
	`SHazaM`	0.731 (0.002)	0.507 (0.002)
	logistic	0.611 (0.002)	0.596 (0.001)
`SHMulate`	`samm`	0.478 (0.001)	0.689 (0.001)
	`SHazaM`	0.489 (0.001)	0.690 (0.001)
	logistic	0.499 (0.002)	0.677 (0.001)

number of mutations per sequence, and the data is more informative for finding motifs with high mutation rates rather than those with low mutation rates. Based on results from Section 3.1, we expect the bias of `samm` to shrink as the number of training observations increases.

4. Data analysis. We fit models to the BCR sequence data obtained from a vaccination study of four transgenic mice published in (Cui et al. (2016)). In this experimental setting the substitutions present in the κ-light chain sequences are unlikely to be affected by natural selection on BCR function. Thus, we restrict our



(a) Survival simulation (b) `SHMulate` simulation

FIG. 4. *Boxplots of the differences between median-centered fitted and true* $\boldsymbol{\theta}$ *values for samm* (*left*), `SHazaM` (*middle*) *and logistic regression* (*right*).

TABLE 2

Statistics of processed κ-light chain data from Cui et al. (2016). `SHazaM` uses all sequences while `samm` samples a single sequence from each clonal family. We filter sequences with indels in all analyses. There are fewer clonal families in the sampled sequences as `samm` filters out sequences with no mutations

	All sequences	Sampled sequences
Number of mutated sequences	15,025	2429
Number of clonal families	2565	2429
Median mutated sequence length	282	282
Average mutation frequency (%)	2.32	2.17
Number of 5-mers in naïve sequences	1014	967

analysis to only κ-light chain data in order to estimate somatic hypermutation rates rather than a combination of somatic hypermutation and selection (Yaari, Uduman and Kleinstein (2012), McCoy et al. (2015), Yaari et al. (2015)). A single naïve sequence can give rise to many different B-cell receptors by somatic hypermutation, forming a so-called "clonal family" which may have varying levels of shared evolutionary history. We use `partis` (Ralph and Matsen IV (2016a)) to assign mutated sequences to clonal families and infer the most likely naïve sequence in each family. In both the sequencing and the clonal family inference there is the possibility of error propagation; we begin our analysis by assuming BCRs are accurately sequenced and assigned to clonal families. The resulting data has the composition shown in Table 2. To mitigate double-counting mutations, we sample a single mutated sequence from each clonal family. Though this discards a lot of the data, we believe this gives more accurate estimates than other approaches that try to use all the data or estimate mutation history; we analyze this issue in more depth in Section C.1 in the Supplementary Material (Feng et al. (2019)).

We fit a 3,5-mer model using `samm` using the same settings as before (Figure 5) though with 5-fold cross-validation to determine the optimal parameter support. The $\boldsymbol{\theta}$ estimate has a block-like and 4-fold-repetitive pattern because many 5-mer motifs were zeroed out during the lasso step. The 95% uncertainty intervals suggest that many motifs have a marked nonzero effect.

Our model recovers many of the well-known "hot" (more mutable) and "cold" spots (less mutable k-mers) which are denoted by the red, blue and green bars in Figure 5. Hot/cold motifs are typically denoted with an underline indicating which position is mutating and represented by degenerate bases $\mathtt{W} = \{\mathtt{A}, \mathtt{T}\}$, $\mathtt{R} = \{\mathtt{A}, \mathtt{G}\}$, $\mathtt{Y} = \{\mathtt{C}, \mathtt{T}\}$, $\mathtt{S} = \{\mathtt{C}, \mathtt{G}\}$, $\mathtt{N} = \{\mathtt{A}, \mathtt{G}, \mathtt{C}, \mathtt{T}\}$. We confirm that many highly mutable 5-mer motifs match the classical hot spot motif $\mathtt{WR\underline{C}}$ and its reverse complement $\mathtt{\underline{G}YW}$ (since the mutation process can happen on either DNA strand) (Rogozin and Diaz (2004)). We also confirm that many less mutable 5-mer motifs match the canonical cold spot $\mathtt{SY\underline{C}/\underline{G}RS}$ (Yaari et al. (2013)). For example, one of the 5-mers we estimate to have high mutability ($\theta = 1.688$) is $\mathtt{AA\underline{G}CT}$, which is of the

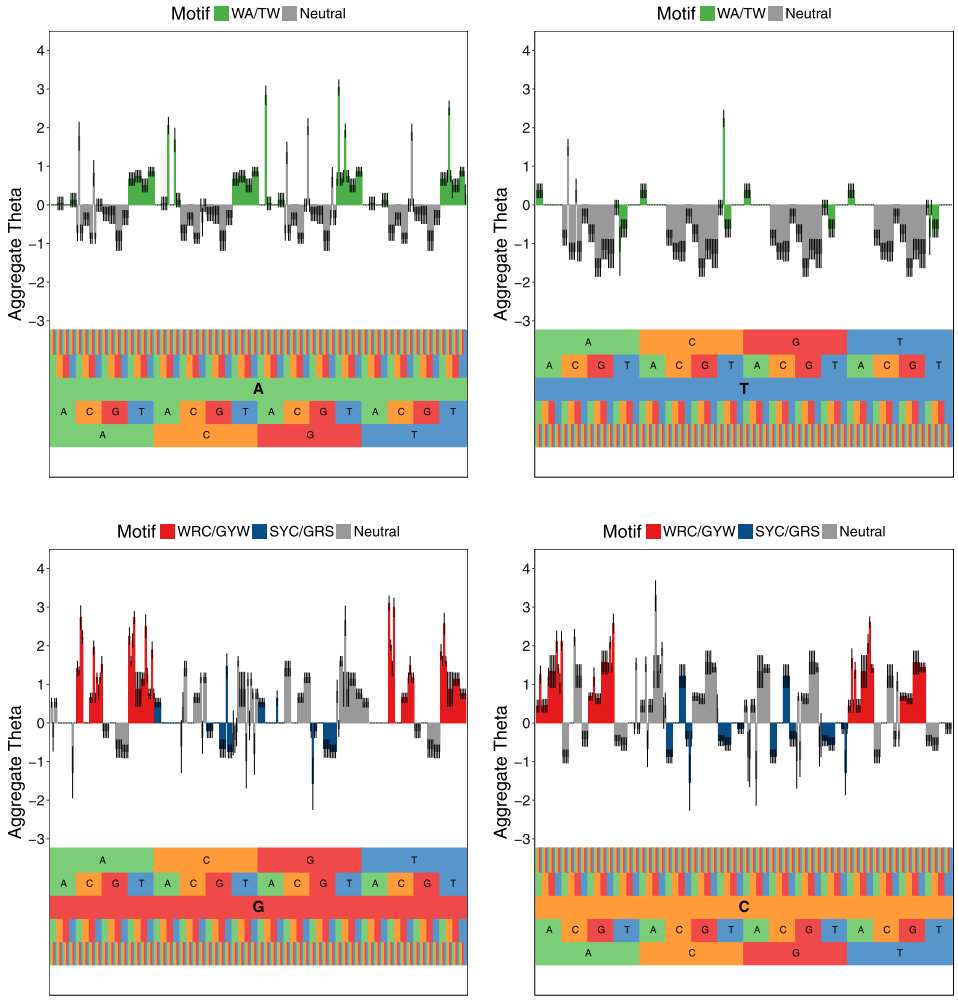


FIG. 5. *Estimated somatic hypermutation model for mouse light chains using* `samm` *for* 5*-mer motifs centered on the bases* `A` *(top left),* `T` *(top right),* `G` *(bottom left) and* `C` *(bottom right). The motif corresponding to an* x*-axis position can be read from bottom to top. Plots depict the estimated aggregate* θ *of* 5*-mer motifs after estimating the model for a* 3,5*-mer model and aggregating estimates using the procedure outlined in Section* 2.5*. A negative value means a reduced mutation rate relative to the baseline hazard, whereas a positive means an enhancement. Well-known hot spots,* `WRC`/`GYW` *and* `WA`/`TW`*, are colored red and green respectively. The well-known cold spot* `SYC`/`GRS` *is colored blue. All other motifs are colored grey. The* 95% *uncertainty intervals for the estimates are depicted by black lines in the center of each bar.*

form `NNGYW` and ends with the 3-mer `GYW`. As `C` is an example of a `Y` nucleotide and `T` is an example of a `W`, `AAGCT` is an example of the hot spot motif `GYW`.

Our model also reveals shortcomings with the current hot and cold spot definitions. Our estimates show significant variability in the mutabilities of motifs, even

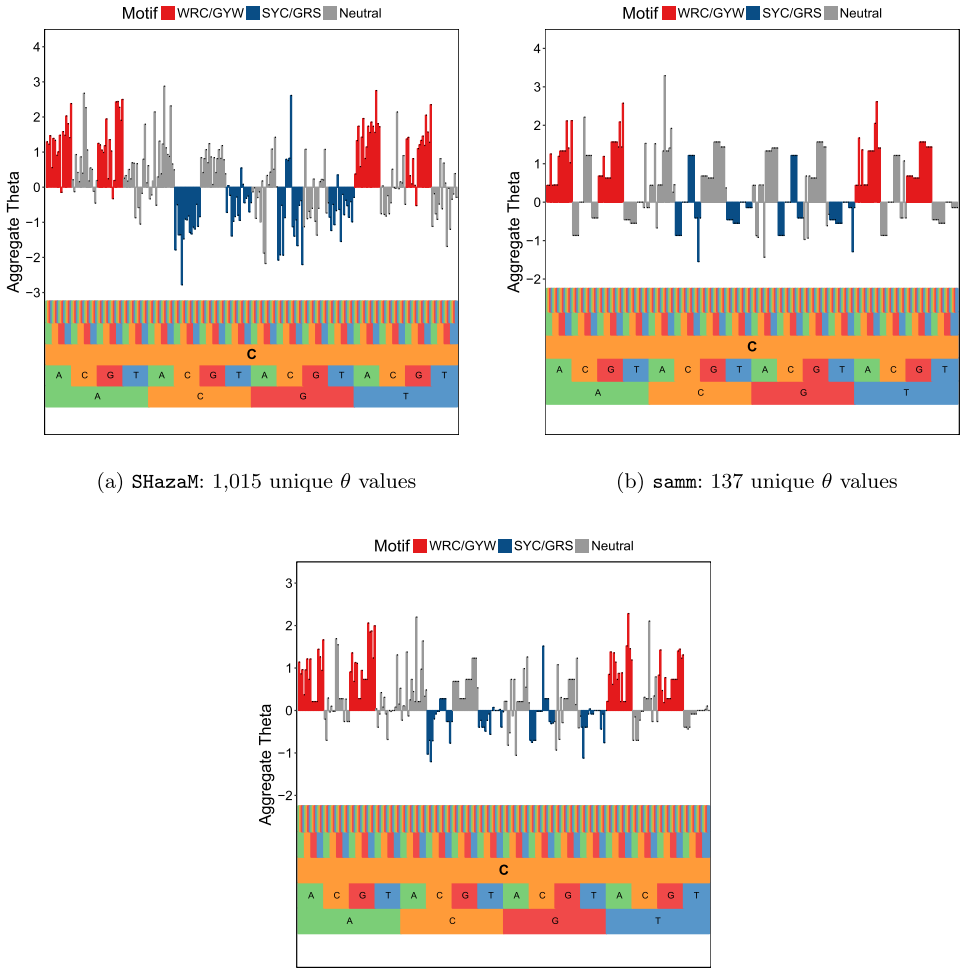
if they contain the same hot or cold spot motif. For instance, in the established literature the ATGGC motif is considered to be a cold spot since it is of the form GRS. We estimate its θ value to be very large ($\theta = 2.206$) relative to the other θ values, suggesting that it is actually a hot spot. We also see SHazaM estimates all motifs of the form CCCNN to have negative mutability, and these are examples of the known cold spot SYC. Estimates from samm show CCCGN has a positive mutability even though it is also of the form SYC, indicating the inner 3-mer CCG may increase mutation rate more than the two C nucleotides to the left of the mutating position. In addition the classic hot spots with a central T nucleotide actually had very low mutability estimates; this suggests that using the well-known WA/TW to identify hot spots may not be appropriate.

Finally, our model suggests that samm can be used to discover new hot and cold spots. For example, consider motifs with the central base C mutating. We find that the mutabilities of the 5-mer CACGC and of the 3-mers GCG, GCT, ACT and ACG are all higher than any motif of the form WRC. As each of these motifs are of the form NRC, this indicates the R nucleotide immediately preceding the mutating C may affect mutation rate more than the W nucleotide two bases away. A well-defined inferential procedure to determine significant collections of hot and cold spots with ample support from the data will require additional future work.

For comparison we fit SHazaM on the same data without sampling a single sequence from each clonal family as was done by Yaari et al. (2013). We also fit the logistic model on the same data as samm. All models use the data to determine the degrees of freedom to use in fitting $\boldsymbol{\theta}$, resulting in the number of unique $\boldsymbol{\theta}$ values fit to be less than the saturated model size of 1024 for a 5-mer model. SHazaM estimated 1015 unique θ values out of a maximum of 1024 while samm only estimated 137 unique θ values and logistic estimated 485. Visually, estimates from the three models look similar with similar hot- and cold-spots, though SHazaM is more "spiky" than samm and logistic (Figure 6). In terms of model interpretability, samm or logistic regression seem to be preferable to SHazaM as they produce much more parsimonious models. The logistic model seems to fit a model that is intermediate to samm and SHazaM in terms of parameter support.

Ideally, we would be able to compare the different methods in terms of their observed data likelihood on a test set. However due to methodological difficulties and incompatibilities of the methods, we were unable to come up with a concrete way to compare the methods. In particular, SHazaM is not a likelihood-based method. In addition, the observed data likelihood for samm is computationally intractable which makes it difficult to compare to other likelihood-based methods. We hope to come up with a good solution for assessing samm on real-world data in the future.

5. Discussion. We have modeled somatic hypermutation of BCR sequences using Cox proportional hazards. Due to the context-dependence of mutation rates, we must take into account the unknown mutation order to compute the full likelihood. To deal with this missing data, we used MCEM, where we marginalize



(a) SHazaM: 1,015 unique θ values

(b) samm: 137 unique θ values

(c) logistic: 485 unique θ values

FIG. 6. *A comparison of fitted aggregate θ values from* SHazaM *(left),* samm *(middle) and logistic regression (right) for* 5*-mer motifs with central base* C*. The* samm *fit is the same as in Figure* 5*. Both* samm *and logistic are* 3,5*-mer fits aggregated into* 5*-mer models.* samm *and logistic tend to fit more parsimonious models compared to* SHazaM*, so the left plot looks more "spiky" than the middle and right ones.* samm *produces the most parsimonious fits among the three methods.*

over the possible mutation orders using Markov chain Monte Carlo. Unlike current methods, our regression framework can model the effect of arbitrary features, such as varying motif lengths and sequence positions. In this paper we use the lasso to perform feature selection and stabilize our estimates in high-dimensional settings. One can easily extend this approach to use other sparsity-inducing penalties to reflect other prior beliefs about the model structure. We show that samm achieves

better performance than the state-of-the-art method under a variety of simulation settings.

There are a few limitations with our current method. We currently subsample our data significantly to ensure our training set is composed of independent observations. This would not be necessary if we were able to perform accurate phylogenetic ancestral sequence estimation using context-sensitive models. In addition our method returns "uncertainty" intervals rather than confidence intervals since there are no guarantees on their nominal coverage. Simulations show that our uncertainty intervals are close to their nominal coverage levels if there is a sufficient amount of data (Figure 3), but better methods may be available.

While the present analysis only considers sequence context, other biologically-motivated features may be just as informative—nucleotide position, proximity to other contexts, etc. By incorporating other types of features into the model, we may be able to help verify or find problems with the currently accepted model of somatic hypermutation (Methot and Di Noia (2017)).

Finally, our model can be used in other contexts to model other biological processes. For instance, our method could be used to model the rate of single-nucleotide polymorphisms (Aggarwala and Voight (2016)) and transcription-factor binding (Zhou and Liu (2004)).

Acknowledgments. We are grateful to Duncan Ralph for assistance performing sequence annotation, clustering and simulating germline repertoires. We would like to thank the Kleinstein lab for generously sharing DNA sequences and especially Jason Vander Heiden for providing us with preprocessed versions of their sequence data.

SUPPLEMENTARY MATERIAL

Supplement to "Survival analysis of DNA mutation motifs with penalized proportional hazards" (DOI: 10.1214/18-AOAS1233SUPP; .pdf). Proofs, data processing details, and additional simulation results.

REFERENCES

AGGARWALA, V. and VOIGHT, B. F. (2016). An expanded sequence context model broadly explains variability in polymorphism levels across the human genome. *Nat. Genet.* **48** 349–355.

BECK, A. and TEBOULLE, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2** 183–202. MR2486527

CAFFO, B. S., JANK, W. and JONES, G. L. (2005). Ascent-based Monte Carlo expectation-maximization. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **67** 235–251. MR2137323

CHAHWAN, R., EDELMANN, W., SCHARFF, M. D. and ROA, S. (2012). AIDing antibody diversity by error-prone mismatch repair. *Semin. Immunol.* **24** 293–300.

COHEN, R. M., KLEINSTEIN, S. H. and LOUZOUN, Y. (2011). Somatic hypermutation targeting is influenced by location within the immunoglobulin V region. *Mol. Immunol.* **48** 1477–1483.

COWELL, L. G. and KEPLER, T. B. (2000). The nucleotide-replacement spectrum under somatic hypermutation exhibits microsequence dependence that is strand-symmetric and distinct from that under germline mutation. *J. Immunol.* **164** 1971–1976.

CUI, A., DI NIRO, R., VANDER HEIDEN, J. A., BRIGGS, A. W., ADAMS, K., GILBERT, T., O'CONNOR, K. C., VIGNEAULT, F., SHLOMCHIK, M. J. et al. (2016). A model of somatic hypermutation targeting in mice based on high-throughput Ig sequencing data. *J. Immunol.* **197** 3566–3574.

DEMPSTER, A. P., LAIRD, N. M. and RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B* **39** 1–38. MR0501537

DEZEURE, R., BÜHLMANN, P., MEIER, L. and MEINSHAUSEN, N. (2015). High-dimensional inference: Confidence intervals, p-values and R-software `hdi`. *Statist. Sci.* **30** 533–558. MR3432840

DUNN-WALTERS, D. K., DOGAN, A., BOURSIER, L., MACDONALD, C. M. and SPENCER, J. (1998). Base-specific sequences that bias somatic hypermutation deduced by analysis of out-of-frame human IgVH genes. *J. Immunol.* **160** 2360–2364.

ELHANATI, Y., SETHNA, Z., MARCOU, Q., CALLAN, C. G. JR, MORA, T. and WALCZAK, A. M. (2015). Inferring processes underlying B-cell repertoire diversity. *Philos. Trans. R. Soc. Lond. B, Biol. Sci.* **370** 20140243.

FENG, J., SHAW, D. A., MININ, V. N., SIMON, N. and MATSEN IV, F. A. (2019). Supplement to "Survival analysis of DNA mutation motifs with penalized proportional hazards." DOI:10.1214/18-AOAS1233SUPP.

GOGGINS, W. B., FINKELSTEIN, D. M., SCHOENFELD, D. A. and ZASLAVSKY, A. M. (1998). A Markov chain Monte Carlo EM algorithm for analyzing interval-censored data under the Cox proportional hazards model. *Biometrics* **54** 1498–1507.

GUPTA, N. T., VANDER HEIDEN, J. A., UDUMAN, M., GADALA-MARIA, D., YAARI, G. and KLEINSTEIN, S. H. (2015). Change-O: A toolkit for analyzing large-scale B cell immunoglobulin repertoire sequencing data. *Bioinformatics* **31** 3356–3358.

HAYNES, B. F., KELSOE, G., HARRISON, S. C. and KEPLER, T. B. (2012). B-cell-lineage immunogen design in vaccine development with HIV-1 as a case study. *Nat. Biotechnol.* **30** 423–433.

HE, L., SOK, D., AZADNIA, P., HSUEH, J., LANDAIS, E., SIMEK, M., KOFF, W. C., POIGNARD, P., BURTON, D. R. et al. (2014). Toward a more accurate view of human B-cell repertoire by next-generation sequencing, unbiased repertoire capture and single-molecule barcoding. *Sci. Rep.* **4** 6778.

HERSHBERG, U., UDUMAN, M., SHLOMCHIK, M. J. and KLEINSTEIN, S. H. (2008). Improved methods for detecting selection by mutation analysis of Ig V region sequences. *Int. Immunol.* **20** 683–694.

HESTERBERG, T., CHOI, N. H., MEIER, L. and FRALEY, C. (2008). Least angle and l_1 penalized regression: A review. *Stat. Surv.* **2** 61–93. MR2520981

HOBOLTH, A. (2008). A Markov chain Monte Carlo expectation maximization algorithm for statistical analysis of DNA sequence evolution with neighbor-dependent substitution rates. *J. Comput. Graph. Statist.* **17** 138–162. MR2424799

HOEHN, K. B., LUNTER, G. and PYBUS, O. G. (2017). A phylogenetic codon substitution model for antibody lineages. *Genetics* **206** 417–427.

HWANG, D. G. and GREEN, P. (2004). Bayesian Markov chain Monte Carlo sequence analysis reveals varying neutral substitution patterns in mammalian evolution. *Proc. Natl. Acad. Sci. USA* **101** 13994–14001.

HWANG, J. K., WANG, C., DU, Z., MEYERS, R. M., KEPLER, T. B., NEUBERG, D., KWONG, P. D., MASCOLA, J. R., JOYCE, M. G. et al. (2017). Sequence intrinsic somatic mutation mechanisms contribute to affinity maturation of VRC01-class HIV-1 broadly neutralizing antibodies. *Proc. Natl. Acad. Sci. USA* **114** 8614–8619.

KALBFLEISCH, J. D. and PRENTICE, R. L. (2011). *The Statistical Analysis of Failure Time Data*. *Wiley Series in Probability and Mathematical Statistics* **360**. Wiley, New York. MR0570114

LEEB, H., PÖTSCHER, B. M. and EWALD, K. (2015). On various confidence intervals post-model-selection. *Statist. Sci.* **30** 216–227. MR3353104

LEFRANC, M.-P. (2014). Immunoglobulins: 25 years of immunoinformatics and IMGT-ONTOLOGY. *Biomolecules* **4** 1102–1139.

LEFRANC, M.-P., GIUDICELLI, V., GINESTOUX, C., BODMER, J., MÜLLER, W., BONTROP, R., LEMAITRE, M., MALIK, A., BARBIÉ, V. et al. (1999). IMGT, the international ImMunoGeneTics database. *Nucleic Acids Res.* **27** 209–212.

LOUIS, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *J. Roy. Statist. Soc. Ser. B* **44** 226–233. MR0676213

MCCOY, C. O., BEDFORD, T., MININ, V. N., BRADLEY, P., ROBINS, H. and MATSEN, F. A. IV (2015). Quantifying evolutionary constraints on B-cell affinity maturation. *Philos. Trans. R. Soc. Lond. B, Biol. Sci.* **370** 20140244.

METHOT, S. P. and DI NOIA, J. M. (2017). Chapter two—Molecular mechanisms of somatic hypermutation and class switch recombination. In *Advances in Immunology* (F. W. Alt, ed.) **133** 37–87. Academic Press, San Diego, CA.

NESTEROV, Y. (2013). Gradient methods for minimizing composite functions. *Math. Program.* **140** 125–161. MR3071865

PHAM, P., BRANSTEITTER, R., PETRUSKA, J. and GOODMAN, M. F. (2003). Processive AID-catalysed cytosine deamination on single-stranded DNA simulates somatic hypermutation. *Nature* **424** 103–107.

RALPH, D. K. and MATSEN IV, F. A. (2016a). Consistency of VDJ rearrangement and substitution parameters enables accurate B cell receptor sequence annotation. *PLoS Comput. Biol.* **12** 1–25.

RALPH, D. K. and MATSEN IV, F. A. (2016b). Likelihood-based inference of B cell clonal families. *PLoS Comput. Biol.* **12** e1005086.

ROGOZIN, I. B. and DIAZ, M. (2004). Cutting edge: DGYW/WRCH is a better predictor of mutability at G: C bases in Ig hypermutation than the widely accepted RGYW/WRCY motif and probably reflects a two-step Activation-Induced Cytidine Deaminase-triggered process. *J. Immunol.* **172** 3382–3384.

ROGOZIN, I. B. and KOLCHANOV, N. A. (1992). Somatic hypermutagenesis in immunoglobulin genes. II. Influence of neighbouring base sequences on mutagenesis. *Biochim. Biophys. Acta* **1171** 11–18.

ROGOZIN, I. B., PAVLOV, Y. I., BEBENEK, K., MATSUDA, T. and KUNKEL, T. A. (2001). Somatic mutation hotspots correlate with DNA polymerase η error spectrum. *Nat. Immunol.* **2** 530–536.

SCHATZ, D. G. and JI, Y. (2011). Recombination centres and the orchestration of V (D) J recombination. *Nat. Rev., Immunol.* **11** 251–263.

SHENG, Z., SCHRAMM, C. A., KONG, R., NISC COMPARATIVE SEQUENCING PROGRAM, MULLIKIN, J. C., MASCOLA, J. R., KWONG, P. D. and SHAPIRO, L. (2017). Gene-specific substitution profiles describe the types and frequencies of amino acid changes during antibody somatic hypermutation. *Front. Immunol.* **8** 537.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* **58** 267–288. MR1379242

TIBSHIRANI, R. et al. (1997). The lasso method for variable selection in the Cox model. *Stat. Med.* **16** 385–395.

TONEGAWA, S. (1983). Somatic generation of antibody diversity. *Nature* **302** 575–581.

UDUMAN, M., YAARI, G., HERSHBERG, U., STERN, J. A., SHLOMCHIK, M. J. and KLEINSTEIN, S. H. (2011). Detecting selection in immunoglobulin sequences. *Nucleic Acids Res.* **39** W499–W504.

WEI, G. C. and TANNER, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J. Amer. Statist. Assoc.* **85** 699–704.

WIEHE, K., BRADLEY, T., RYAN MEYERHOFF, R., HART, C., WILLIAMS, W. B., EASTERHOFF, D., FAISON, W. J., KEPLER, T. B., SAUNDERS, K. O. et al. (2018). Functional relevance of improbable antibody mutations for HIV broadly neutralizing antibody development. *Cell Host Microbe* **23** 759–765.

YAARI, G. and KLEINSTEIN, S. H. (2015). Practical guidelines for B-cell receptor repertoire sequencing analysis. *Gen. Med.* **7** 121.

YAARI, G., UDUMAN, M. and KLEINSTEIN, S. H. (2012). Quantifying selection in high-throughput immunoglobulin sequencing data sets. *Nucleic Acids Res.* **40** e134.

YAARI, G., VANDER HEIDEN, J. A., UDUMAN, M., GADALA-MARIA, D., GUPTA, N., STERN, J. N. H., O'CONNOR, K. C., HAFLER, D. A., LASERSON, U. et al. (2013). Models of somatic hypermutation targeting and substitution based on synonymous mutations from high-throughput immunoglobulin sequencing data. *Front. Immunol.* **4** 358.

YAARI, G., BENICHOU, J. I. C., VANDER HEIDEN, J. A., KLEINSTEIN, S. H. and LOUZOUN, Y. (2015). The mutation patterns in B-cell immunoglobulin receptors reflect the influence of selection acting at multiple time-scales. *Philos. Trans. R. Soc. Lond. B, Biol. Sci.* **370** 20140242.

YEAP, L.-S., HWANG, J. K., DU, Z., MEYERS, R. M., MENG, F.-L., JAKUBAUSKAITĖ, A., LIU, M., MANI, V., NEUBERG, D. et al. (2015). Sequence-intrinsic mechanisms that target AID mutational outcomes on antibody genes. *Cell* **163** 1124–1137.

ZHAO, S., SHOJAIE, A. and WITTEN, D. (2017). In defense of the indefensible: A very naive approach to high-dimensional inference. Preprint. Available ar arXiv:1705.05543.

ZHOU, Q. and LIU, J. S. (2004). Modeling within-motif dependence for transcription factor binding site predictions. *Bioinformatics* **20** 909–916.

J. FENG
N. SIMON
DEPARTMENT OF BIOSTATISTICS
UNIVERSITY OF WASHINGTON
SEATTLE, WASHINGTON 98195
USA
E-MAIL: nrsimon@u.washington.edu

D. A. SHAW
F. A. MATSEN IV
COMPUTATIONAL BIOLOGY PROGRAM
FRED HUTCHINSON CANCER RESEARCH CENTER
SEATTLE, WASHINGTON 98109
USA
E-MAIL: matsen@fredhutch.org

V. N. MININ
DEPARTMENT OF STATISTICS
UNIVERSITY OF CALIFORNIA, IRVINE
IRVINE, CALIFORNIA 92697
USA
E-MAIL: vminin@uci.edu