PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Predicting B cell receptor substitution profiles using public repertoire data

Amrit Dhar[1,2☯], **Kristian Davidsen**[2☯], **Frederick A. Matsen IV**[2*], **Vladimir N. Minin**[3*]

1 Department of Statistics, University of Washington, Seattle, Washington, United States of America, **2** Computational Biology, Fred Hutchinson Cancer Research Center, Seattle, Washington, United States of America, **3** Department of Statistics, University of California, Irvine, California, United States of America

☯ These authors contributed equally to this work.
* matsen@fredhutch.org (FAM); vminin@uci.edu (VNM)

OPEN ACCESS

Citation: Dhar A, Davidsen K, Matsen FA, IV, Minin VN (2018) Predicting B cell receptor substitution profiles using public repertoire data. PLoS Comput Biol 14(10): e1006388. https://doi.org/10.1371/journal.pcbi.1006388

Editor: Aleksandra M. Walczak, CNRS and Ecole Normale Superieure, FRANCE

Received: February 20, 2018

Accepted: July 22, 2018

Published: October 17, 2018

Data Availability Statement: All data available at https://github.com/krdav/SPURF.

Funding: FAM was supported by the National Institutes of Health (https://www.nih.gov) grants R01 GM113246, R01 AI12096, and U19 AI117891. FAM was supported in part by a Faculty Scholar grant from the Howard Hughes Medical Institute (http://www.hhmi.org) and the Simons Foundation (https://www.simonsfoundation.org). AD was supported by the National Science Foundation (https://www.nih.gov) IGERT DGE-1258485. The funders had no role in study design, data collection

Abstract

B cells develop high affinity receptors during the course of affinity maturation, a cyclic process of mutation and selection. At the end of affinity maturation, a number of cells sharing the same ancestor (i.e. in the same "clonal family") are released from the germinal center; their amino acid frequency profile reflects the allowed and disallowed substitutions at each position. These clonal-family-specific frequency profiles, called "substitution profiles", are useful for studying the course of affinity maturation as well as for antibody engineering purposes. However, most often only a single sequence is recovered from each clonal family in a sequencing experiment, making it impossible to construct a clonal-family-specific substitution profile. Given the public release of many high-quality large B cell receptor datasets, one may ask whether it is possible to use such data in a prediction model for clonal-family-specific substitution profiles. In this paper, we present the method "Substitution Profiles Using Related Families" (SPURF), a penalized tensor regression framework that integrates information from a rich assemblage of datasets to predict the clonal-family-specific substitution profile for any single input sequence. Using this framework, we show that substitution profiles from similar clonal families can be leveraged together with simulated substitution profiles and germline gene sequence information to improve prediction. We fit this model on a large public dataset and validate the robustness of our approach on two external datasets. Furthermore, we provide a command-line tool in an open-source software package (https://github.com/krdav/SPURF) implementing these ideas and providing easy prediction using our pre-fit models.

Author summary

Antibody engineering can be greatly informed by knowledge about the underlying affinity maturation process. As such this can be probed by sequencing, but unfortunately, in practice often only one member of the clonal family is sequenced, making it difficult to determine a set of possible amino acid mutations that would retain the original antibody antigen binding affinity. We overcome this data sparsity by developing a statistical learning approach that leverages vast information about amino acid preferences available in

and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

public immune system repertoire data. We use a penalized regression approach to devise a flexible statistical model that integrates multiple sources of information into a coherent prediction framework and validate our prediction algorithm using subsampling and held out data.

Introduction

In the therapeutic antibody discovery and engineering field, researchers commonly isolate antibodies from animal or human immunizations and screen for functional properties such as binding to a target protein. Following the initial screening process, a small number of well-behaving antibodies (hits) are isolated for more rigorous examination of their biophysical properties in order to determine their potential as a therapeutic. After this stage, only a few final antibodies remain as lead candidates. However, even these carefully selected antibodies often have immunogenic peptides or other undesirable properties such as poor thermo/chemical stability and aggregation tendencies. To address these problems, the art of antibody engineering has emerged [1], with numerous rational design strategies developed to mitigate aggregation. Researchers have removed hydrophobic surface patches to avoid aggregation [2–5], "deimmunized" complementarity-determining regions by screening immunogenic peptides and mutating positions detrimental for peptide MHCII binding [6], and improved thermostability through stable framework grafting [7] and targeted mutagenesis using predictions from proprietary structure/sequence analysis software [8]. Although referred to as "rational", the choice of which amino acid to use for a site-directed mutation is often made using 1) the germline as a reference, 2) biochemical similarity between amino acids, or 3) the highest probability amino acid from a generic substitution matrix (e.g. BLOSUM) [9]. However, neither of these three methods are explicitly designed to conserve antibody functionality (i.e. binding to the same epitope with the same kinetics), so mutations are likely to have negative side effects on affinity. These considerations motivate a prediction problem: given a B cell receptor (BCR) sequence, which positions can be modified, and to which amino acids, without drastically changing the binding properties of the resulting BCR?

An immunization-derived antibody has already implicitly explored the mutational space through the population of B cells sharing the same naive ancestor, referred to as its clonal family (CF). The members of a CF arise during affinity maturation in a germinal center and carry fitness information about the effect of amino acid substitutions. A profile of the observed substitutions aggregated over all the B cells in a CF reveals which sites are more conserved, which sites can be more freely edited, and which amino acids can be used for replacements. However, we generally do not sequence all the B cells that are released from a germinal center so the information to make such a substitution profile is lost. Thus, we can formulate a more specific version of our prediction problem: given bulk BCR data and a single input sequence, can we infer the most likely per-site substitutions that are allowed in its true germinal center clonal family?

We begin by reviewing the natural mutation and selection process of germinal center affinity maturation. The Darwinian selection undertaken inside a germinal center is driven by B cells' ability to bind the antigen through the membrane-embedded BCR. The highly-mutated population of B cells in a germinal center is under stringent selection, driving the cell population towards higher and higher affinity until the germinal center is dissolved. Each germinal center is seeded by around one hundred naive B cells, but eventually internal competition makes one or a few of these lineages take over the whole germinal center [10]. Although B cells

in the germinal center reaction experience an extraordinarily high mutation rate (10^6 fold higher than the regular somatic mutation rate [11]), they rarely harbor more than 15% mutations at the DNA level [12]. However, since they must maintain some degree of antigen specificity to survive during the course of the germinal center reaction, lineages evolve in small incremental steps [13, 14] and therefore, even lineages that drift far away from their naive B cell ancestor most likely maintain the same epitope specificity throughout the germinal center reaction [15].

We can describe the combination of germinal center mutation and selection dynamics by computing per-site amino acid frequency vectors from observed BCR sequence data. We follow previous authors in calling site-specific amino acid probability vectors "substitution profiles", where each vector in a profile stores the probabilities of observing the 20 different amino acids at a given site [16]. We use the concept of a clonal family, defined by a shared heavy chain inferred naive DNA sequence, to segment BCR sequences into evolutionarily-related groups [17]; some practitioners refer to these groups as lineages. CF inference is highly informed by nucleotide sequences and therefore performed using DNA sequences. This makes DNA-level information necessary even though germinal center selection operates at the protein level and synonymous codons do not possess any fitness advantages (modulo transcription rate differences and codon bias, which we follow many others in ignoring here). The per-site amino acid frequency vectors described above form the substitution profile estimates; the substitution profile estimates converge to the true substitution profiles as the number of sequences sampled from the same CF tends to infinity.

Most CFs do not contain enough sequences in order to get a detailed substitution profile estimate. Indeed, most CFs in repertoire sequencing (Rep-Seq) samples have few members and a large fraction are singletons due to the exponential nature of the CF size distribution [17]. Additionally, many antibody screening methods are not geared towards whole repertoire sequencing. One may wish, then, to enhance the substitution profile estimates for data-sparse CFs with substitution profile information from similar CFs.

In this paper, we present "Substitution Profiles Using Related Families" (SPURF), a penalized tensor regression framework that integrates multiple sources of information to predict the CF-specific amino acid frequency profile for a single input BCR sequence (Fig 1). Some of these information sources include substitution profiles for CFs in large, publicly available BCR sequence datasets and germline gene sequence information. We combine the local context-specific profile information with global profile information derived from other related germinal centers by regularizing the noisy local substitution profile estimate and pooling it closer towards more robust global profile estimates. Even though each germinal center focuses on binding to a unique epitope context, there are structural and possibly functional properties associated with BCR sequences that are common across germinal centers that we can leverage.

In addition, our inference machinery uses both standard and spatial lasso penalties as model regularizers and, as a result, furnishes sparse, interpretable parameter estimates. While our output type shares some similarities to that described by [16], the proposed objective, approach, and details differ (e.g. they predict substitution profiles for gene families, we predict substitution profiles for CFs). We enable substitution profile prediction for single input BCR sequences based on profiles derived from a high-quality repertoire dataset that contains B cell samples from many human donors. To demonstrate the usefulness of our technique, we validate SPURF on two external datasets—one containing CFs extracted from a single human donor and the other focusing on a single CF of a HIV broadly neutralizing antibody. Lastly, we implement SPURF in an open-source software package (https://github.com/krdav/SPURF), which outputs a predicted CF-specific substitution profile and an associated logo plot based on a single input BCR sequence.

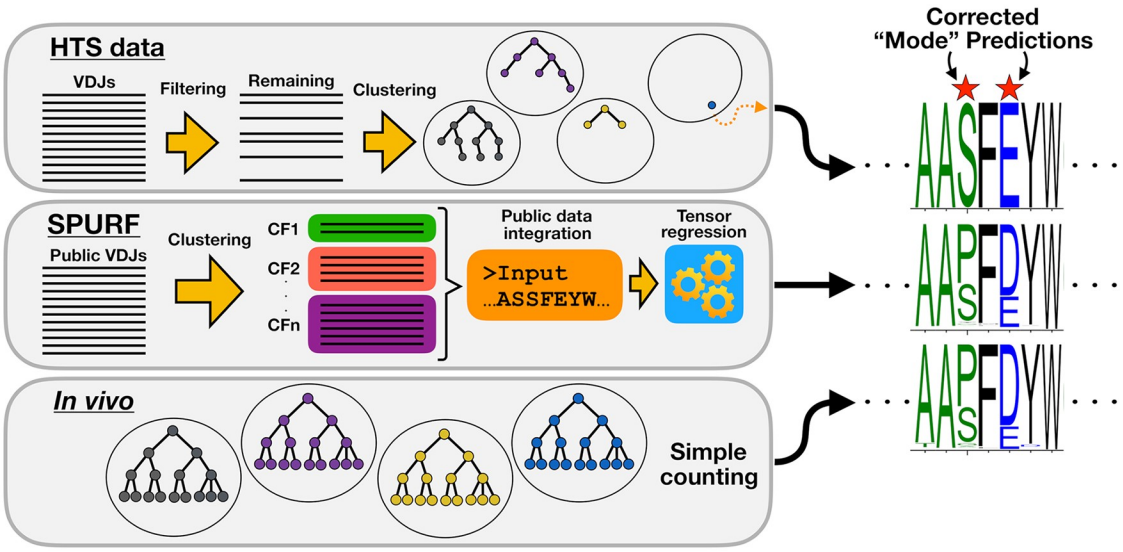

Fig 1. Amino acid substitution profiles viewed from three different perspectives. High-throughput sequencing data (HTS data) yields large amounts of VDJ sequences, but because of uneven sampling many CFs will be sampled just once, resulting in poor representations of the amino acid substitution profiles of those true CFs. "Substitution Profiles Using Related Families" (SPURF) is a statistical framework that integrates large scale Rep-Seq data to predict amino acid substitution profiles for singleton CFs. *In vivo* affinity maturation will test many different mutations and the resulting CFs reflect the amino acid substitution profiles that we attempt to predict.

https://doi.org/10.1371/journal.pcbi.1006388.g001

Methods

Overview

The aim of our model is to take a single sequence and predict the site-wise amino acid frequencies as would be found in the full CF from which this single sequence derived. We will refer to this as the sequence's CF-specific substitution profile. For this prediction problem, we have no direct information about this desired substitution profile other than the information contained in the input sequence itself, but we may use other information (e.g. from the inferred germline gene, simulated substitutions, or information derived from published BCR sequence datasets). For large CFs, a CF-specific substitution profile can be constructed simply by counting and making a per-site frequency matrix, with the rows of the matrix representing each of the 20 amino acids, and the columns being the sequence positions.

For training, we extract a collection of such large CFs and use them to build "ground truth" CF-specific substitution profiles as a training set for fitting the model. A randomly sampled single sequence is then taken out from each of these large CFs to predict the substitution profile, which is compared to the ground truth. We refer to these single sequences, sampled from large CFs, as subsamples.

To make the best possible prediction, we need a flexible model framework that can accommodate different sources of information seamlessly (Fig 2). For example, previous work by [18] and [19] suggests that the various V genes have different characteristic paths of diversification. We can obtain a data-driven summary of that intuition by building profiles from large Rep-Seq data sets stratified by V gene. We may also think that the neutral substitution process is an important factor in determining substitution profiles [18]. We can quantify that sort of information by repeatedly simulating the neutral substitution process using a context-sensitive model [20]. We call each external data set (e.g., V gene alignments and sequences simulated

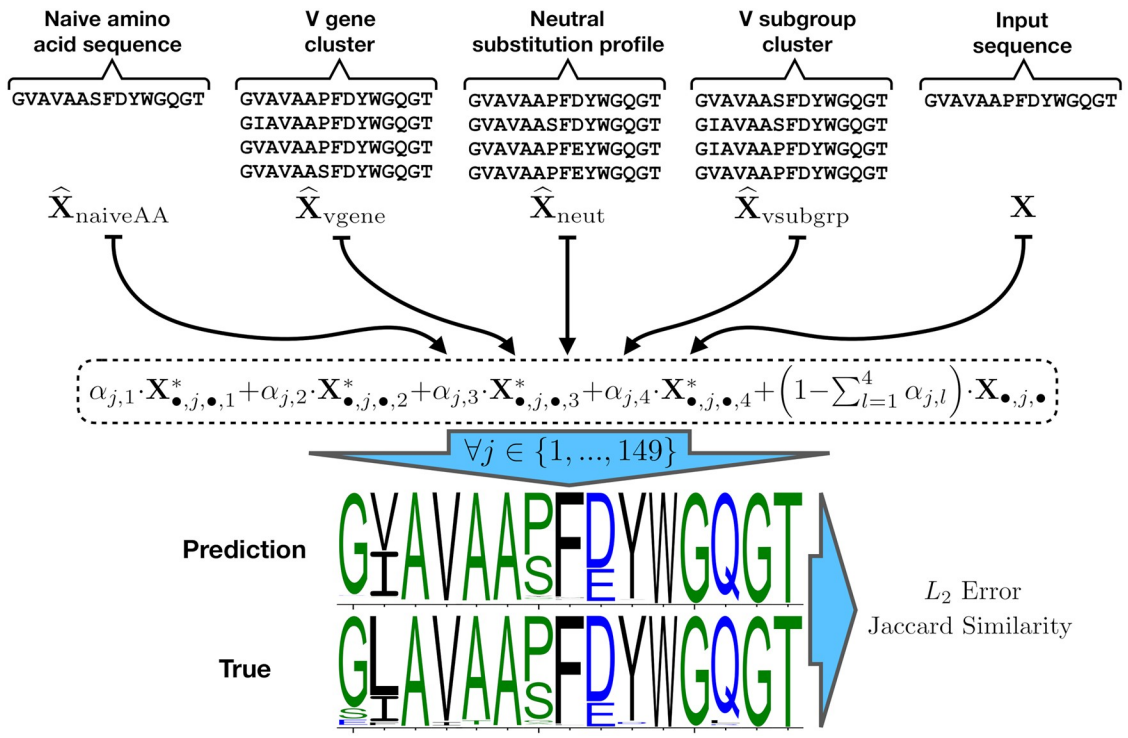

Fig 2. Model overview figure. SPURF uses a per-site linear combination of substitution profiles from diverse sources to predict complete substitution profiles from a single member of a CF. At the top are the different profiles that serve as inputs to the model, some directly related to the naive sequence ($\hat{\mathbf{X}}_{\text{naiveAA}}$ and $\hat{\mathbf{X}}_{\text{neut}}$), and others partitions of the public Rep-Seq datasets ($\hat{\mathbf{X}}_{\text{vgene}}$ and $\hat{\mathbf{X}}_{\text{vsubgrp}}$). To predict a substitution profile, a weighted average is taken over the input sequence $\mathbf{X}$ and external profiles $\mathbf{X}^* = \{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}, \hat{\mathbf{X}}_{\text{neut}}, \hat{\mathbf{X}}_{\text{vsubgrp}}\}$ (see the dashed line bubble). The vertical blue arrow indicates that the weighted average (in the dashed line bubble) occurs at each of the 149 AHo positions. Once a predicted profile is generated, this is compared to ground truth using either L_2 error or Jaccard similarity as a performance metric. The $\boldsymbol{\alpha}$ vectors are estimated by optimizing the objective function, which also includes a statistical regularization term to prevent overfitting.

https://doi.org/10.1371/journal.pcbi.1006388.g002

from the neutral substitution process) that can be used to predict the CF substitution profile of interest a source of profile information.

To make predictions using these types of information, we need a way of describing the various sites, and a way of integrating the information across the sites. We use the AHo numbering scheme [21] to provide a single coordinate system to all sequences via its fixed-length numbering vector going from 1 to 149. Given this coordinate system, we use a site-wise weighted average of the input predictive profiles using a α weight vector for each source of profile information.

To train this model, we fit the $\boldsymbol{\alpha}$ vectors by minimizing some objective function that quantifies the difference between the predicted profiles (where the prediction uses the subsampled sequence and the external profile information) and the "ground truth" substitution profiles from the large CFs. Any objective function could be used, but here we provide implementations of two such functions, a "fine-grained" L_2-error-based objective and a "coarse-grained" Jaccard-similarity-based objective [22].

We use two forms of regularization to avoid overfitting the many parameters of this model. This includes a standard lasso penalty to shrink weights to zero that do not contribute significantly to prediction performance [23]. We also use a fused lasso penalty [24, 25] to smooth differences between parameters at nearby sites in the sequence. These regularization terms have

tuning parameters that regulate the strength of the penalties and are estimated using cross-validation.

Given this setup, a forward stepwise selection procedure is run with cross-validation to pick the set of external profiles to use in the final model. As a last check, this model is tested on two external datasets to give a fair estimate of the prediction performance.

Data

We divide input data into two parts, with each part for a respective purpose: 1) model fitting and model testing and 2) providing "public" substitution profiles over clustered data to be used by our model. Throughout this work, we are careful to not use the same data for both purposes as this would bias our estimates; as a final validation, we test SPURF on two external datasets which are only used in this validation. Because we do not model sequence error, we only include high-quality data that we have high confidence in. We collect post-processed data files from 6 published works on Rep-Seq, which we refer to as repertoire data 1 to 6 (RD1-6):

1. RD1 from [26], which is an Illumina MiSeq re-sequencing of the samples in [27], where they sequence multiple time-points before and after influenza vaccination of 3 donors using the 454 pyrosequencing platform.
2. RD2 from [28], from a study of the auto-immune disease Myasthenia Gravis (MG), in which 9 MG patients and 4 healthy donors participate.
3. RD3 from [29], containing data from different tissues in a study of B cell response in 4 multiple sclerosis patients.
4. RD4 from [30], from a study of neutralizing antibodies against the West Nile virus by sequencing naive and memory cells from 7 virus infected donors.
5. RD5 from [31], from a study of Rep-Seq error correction by sequencing naive, plasma, and memory cells from a single healthy donor.
6. RD6 from [32], from the "B cell tissue atlas" acquired from the ImmuneDB web portal.

All datasets are acquired in their post-processed form with read processing performed as described in their respective publications.

The first five datasets (RD1-5) are prepared from unique molecular identifier (UMI) barcoded cDNA spanning the whole VDJ region and sequenced on the Illumina MiSeq platform using overlapping paired-end reads. Using the UMI, these reads are processed to address both PCR and sequencing errors giving high confidence reads [33]. Briefly, UMIs are used for error correction in conjunction with either of the pRESTO [34] or MIGEC [33] processing pipelines and an appropriate Phred quality score cutoff. Paired-end reads are assembled using pRESTO and only the set of high confidence assembled reads constitute the final dataset used in this work. RD6 is the only dataset not prepared with UMIs; however, it is sequenced directly from genomic DNA (gDNA) instead of the more common practice of sequencing mRNA. Sequencing gDNA has the benefit of avoiding mutations introduced by the transcription machinery as well as mutations introduced in the RT-PCR step. On the other hand, DNA sequencing is not able to discriminate between expressed versus unexpressed BCRs (e.g. in the case of faulty VDJ recombination) and therefore we apply aggressive filtering of non-functional BCR sequences. We prefer quality over quantity and therefore avoid datasets from the 454 technology because of their higher indel frequencies compared to those from Illumina technologies [35].

Individual sequence files are merged based on donor identity so that the number of sample files matches the number of donors; this process yields 33 donor files. The donor files are then

annotated and partitioned into CFs using the `partis` software [17, 36]. Each donor file is run separately from the other files so CFs are defined by their unique `partis`-inferred naive sequence and donor identity. To ensure we obtain the highest quality and most biologically relevant sequences, `partis` is run in its most restrictive mode, discarding all reads with VDJ recombinations that are deemed as unproductive because of out-of-frame N/P junction nucleotides, missing invariant codons, or stop codons inside the VDJ region; furthermore, the most accurate `partis` partitioning mode ("full") is used to get the best CF estimates. Lastly, productive VDJ-recombined sequences are removed if they contain indels to assure concordance between the length of the naive sequence and the length of the read sequences in its CF.

At this stage, some sequences contain ambiguous bases (e.g., because of primer masking); these are allowed to pass only if the ambiguous bases are inside the first or last 30 nucleotides of the VDJ region (equivalent to the length of the potentially masked PCR primers), otherwise they are discarded. This is a way of substituting the error-prone ends with neutral bases that minimize variance and maintain a conservative estimate of the substitutions; we also note that this has no apparent effect on the subsequently-described estimates (Figs 3 and 4). For all sequences that pass this requirement, ambiguous bases are substituted with bases from the naive sequence in batches of 3 nucleotides (i.e. one codon) at a time until all ambiguous bases are resolved. Sequences are then translated into their respective amino acid sequences and de-duplication of repeated amino acid sequences is done within each CF. Because our statistical methodology operates on these amino acid sequences, we use the word "sequence" in subsequent sections to refer to these amino acid sequences. All CFs with fewer than 5 unique sequences are discarded. From these remaining CFs, their inferred naive sequences are used for antibody sequence numbering with the ANARCI software [37] under the AHo numbering scheme [21]. As a result of our restriction to non-indel sequences, all sequences within a given CF have equal length; thus, the AHo numbering from the naive sequence can be positionally transferred to all its CF-related read sequences. Finally, for each CF, the amino acid usage is

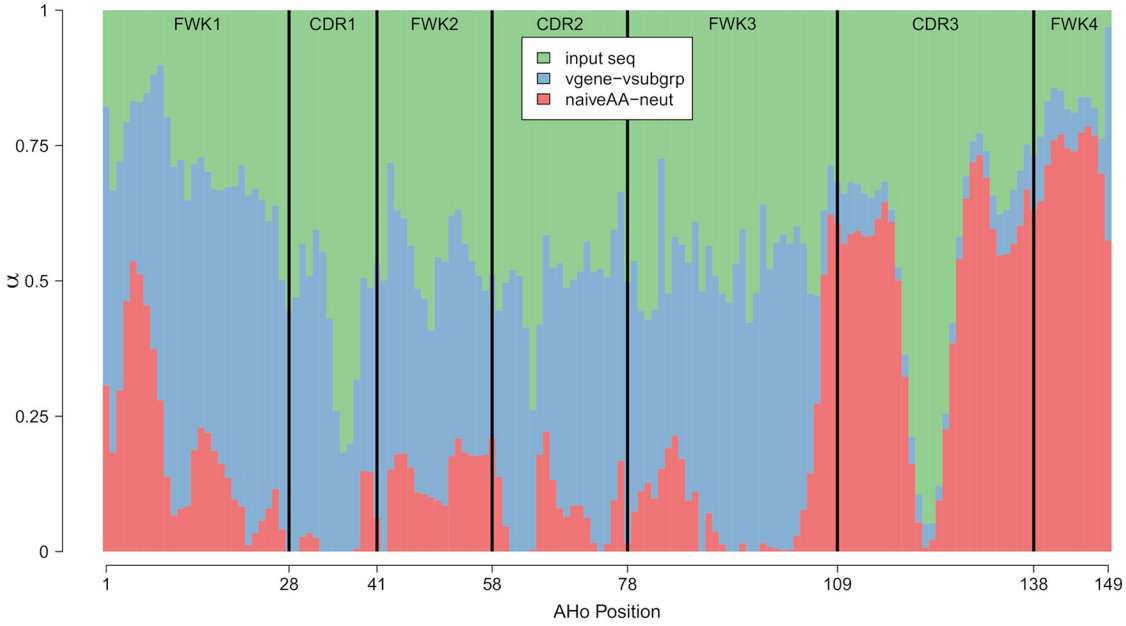

Fig 3. A stacked barplot of the estimated parameter values of α from the best regularized L_2 model. For convenience, we aggregate the estimates of α associated with $\hat{\mathbf{X}}_{\text{vgene}}$ and $\hat{\mathbf{X}}_{\text{vsubgrp}}$ (blue) and with $\hat{\mathbf{X}}_{\text{naiveAA}}$ and $\hat{\mathbf{X}}_{\text{neut}}$ (red). The black vertical lines represent the boundaries between the different CDRs and FWKs.

https://doi.org/10.1371/journal.pcbi.1006388.g003

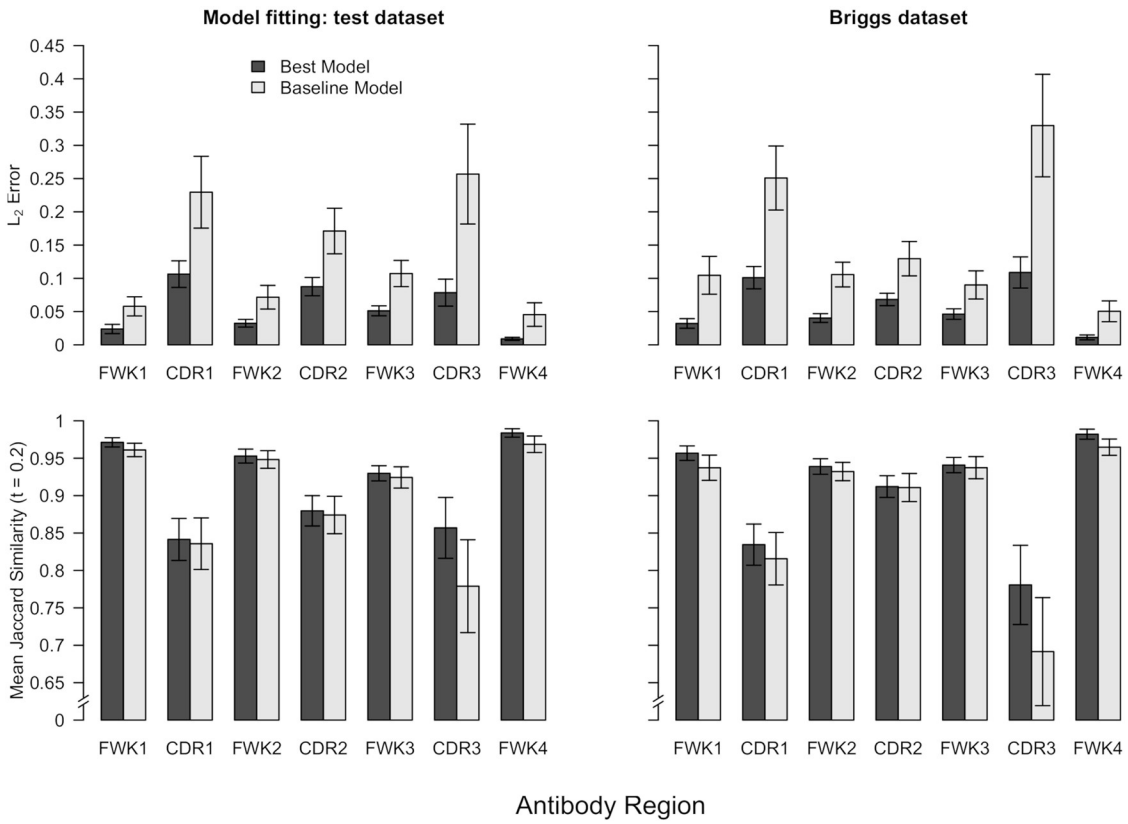

Fig 4. The model performance results across the different antibody regions on the model fitting test dataset and the Briggs validation dataset. In these plots, we compare the performances from our best models to the baseline predictive performances using only the input sequence (i.e. model predictions with all parameter values of $\boldsymbol{\alpha}$ set to 0). The error bars show bootstrap standard errors.

https://doi.org/10.1371/journal.pcbi.1006388.g004

extracted as a vector of counts at each AHo position. This overall dataset, which we call the "aggregated" dataset, contains 518,174 sequences distributed over 31,893 CFs and is built as a matrix of counts with rows denoting CFs and columns representing AHo positions and amino acid identities. All data used to build this aggregated dataset is public and freely available. We provide the data partitioned into CFs and numbered into AHo numbering for download on Zenodo (https://doi.org/10.5281/zenodo.1289984).

Model fitting dataset. To fit our CF-specific substitution profile prediction model, it is desirable to use the CFs from the aggregated dataset with the most sequence members so we can train using the observed substitution profiles with the least amount of noise; on the other hand, it is also desirable to extract CFs from as many donors as possible to avoid overfitting towards a few similar donors. To achieve both goals, we pick 500 CFs as a "model fitting" dataset as follows. We first exclude any CFs with less than 100 sequences from being eligible to be picked. We then cycle through donors, each time picking the largest remaining eligible CF. If a donor does not have any remaining eligible CFs, it is skipped. The process ends when 500 CFs are found; all unpicked CFs are used as the "public" dataset.

In addition, we perform subsampling for each CF in the model fitting dataset; this is the information from which we would like to predict the full profile. First, a single sequence is randomly chosen from each CF, then `partis` is re-run using each of these subsampled sequences to re-do the VDJ annotation and naive sequence inference. For some inferred naive

DNA sequences, a stop codon is incidentally present in the N/P nucleotides of the junction region; these are considered spurious and replaced by the identically positioned codon from the input sequence. We stress that the CF-specific annotation and naive sequence are inferred solely based on the subsampled sequence itself and are not determined using information from the other CF sequence members. Additionally, the parameters used within the `partis` clustering and annotation procedure are derived from an external dataset. Once we finish the `partis` inference process on the subsampled sequences, we construct the amino acid count matrix for these same sequences; we denote these substitution profiles as the "subsampled" profiles because they are subsampled from the "full" profiles in the model fitting dataset.

Simulation of neutral substitution profiles. For each of the 500 subsampled substitution profiles, we also simulate a neutral substitution profile via a context-sensitive model. For each subsampled sequence, we calculate its number of somatic hypermutations (SHMs) and introduce that number of mutations sequentially into the inferred naive DNA sequence according to the BCR-specific neutral substitution model S5F [20]. Once the last mutation is introduced, the simulated DNA sequence is translated into an amino acid sequence and stored as a sample of the neutral substitution process. This procedure is repeated 10,000 times and the count profile aggregated over all the samples is referred to as the "neutral" profile.

External validation datasets. For validation, two test sets are generated: the first called "Briggs", is made from the healthy donor single cell droplet sequencing dataset described in [12]. Briefly, the data is made by passing 3 million B cells into 6 emulsion pools, each droplet with a unique barcode, and then reverse transcribing mRNA inside these droplets, attaching both a droplet and a molecular barcode. After breaking the emulsion, cDNA is sequenced and processed using UMI consensus building using pRESTO. The highest-quality UMI consensus sequence is extracted from each drop and aggregated into the final heavy chain dataset, which is then further partitioned into CFs using `partis`. Finally, this validation dataset is built up in the same manner as the model fitting dataset, where the only difference is that we allow smaller CFs to enter this dataset (minimum 28 sequences; Table 1) in order to increase the number of extracted CFs to 100.

As all the above described datasets are repertoire wide datasets with hundreds of clonal families, we sought to find a suitable dataset with focus on a single large CF. For this, we created the second test set curated from the "Liao" dataset which comes from a well studied broadly neutralizing HIV clone, CH103, described in [38]. To prepare the raw heavy chain sequences from [38], they were annotated and indel reversed by `partis`, following reconstruction of the whole VDJ region by substituting ambiguous bases with bases from the `partis`-inferred naive sequence if necessary. Finally, sequences unable to be annotated within the standard 149

Table 1. Number of donors (N_{donors}), number of CFs (N_{CF}), number of sequences from all CFs (Total N_{seq}), smallest CF size (Min N_{seq}), median CF size (Median N_{seq}), and maximum CF size (Max N_{seq}).

Dataset	Dataset summary statistics					
	N_{donors}	N_{CF}	Total N_{seq}	Min N_{seq}	Median N_{seq}	Max N_{seq}
Aggregated	33	31,893	518,174	5	9	2,709
Model fitting	15	500	98,887	100	147	2,709
Public	33	31,393	419,287	5	8	104
Briggs	1	100	6,702	28	44	370
Liao	1	1	312	312	312	312

"Aggregated" is the base dataset aggregating RD1-6. "Model fitting" refers to the dataset with the 500 largest CFs from the "Aggregated" dataset. "Public" is the dataset left after the "Model fitting" dataset is extracted from the "Aggregated" dataset. "Briggs" and "Liao" are the external validation datasets used for testing.

https://doi.org/10.1371/journal.pcbi.1006388.t001

position AHo numbering scheme were filtered out, leaving 312 sequences (available on the SPURF GitHub repository). For the Liao dataset, the prediction error is measured using all sequences as input samples, contrary to the repertoire datasets where only a single input sequence from each CF is used.

In summary, our datasets span 35 different donors, $\sim$32,000 clonal families and $\sim$500,000 sequences (Table 1). We note that the distribution of VDJ gene usage is non-uniform but that the "Model fitting" and "Public" datasets have very similar V/J gene usage (S4 and S5 Figs). On the other hand, the "Briggs" dataset does have a distinctly different V/J gene usage distribution compared to the other datasets, which we attribute to the fact that it comes from a single donor.

Input data tensor

Before we present our penalized tensor regression model, we first describe how the input data for the model is constructed, building off the data descriptions in the last subsection. Throughout the rest of this section, we assume the count matrices are normalized to frequencies and reorganized into three-dimensional tensors (i.e. arrays) as follows. For any substitution profile tensor $\mathbf{T} = \{T_{i,j,k}\}$, let $T_{i,j,k}$ denote the frequency of the kth amino acid at the jth AHo position for the ith CF; we represent the subsampled, full, and public substitution profile tensors as $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, respectively. Our goal is to use the subsampled profiles $\mathbf{X}$ to predict the corresponding full substitution profiles $\mathbf{Y}$ (i.e. we want to construct a function $F(\mathbf{X})$ such that $F(\mathbf{X}) \approx \mathbf{Y}$). We incorporate information from the public dataset $\mathbf{Z}$ to enhance these predictions. In addition to the subsampled profiles, we use other types of substitution profiles within $F(\mathbf{X})$:

1. Public substitution profiles segmented by the inferred V-subgroup label ($\hat{\mathbf{X}}_{\text{vsubgrp}}$);
2. Public substitution profiles segmented by the inferred V-gene label ($\hat{\mathbf{X}}_{\text{vgene}}$);
3. Inferred naive sequence "substitution profiles" ($\hat{\mathbf{X}}_{\text{naiveAA}}$);
4. Public substitution profiles segmented by the inferred naive sequence ($\hat{\mathbf{X}}_{\text{naiveAA}-\text{clust}}$);
5. Public substitution profiles segmented by the original frequency profiles ($\hat{\mathbf{X}}_{\text{clust}}$);
6. Neutral substitution profiles ($\hat{\mathbf{X}}_{\text{neut}}$).

To compute the external profiles in $\hat{\mathbf{X}}_{\text{vsubgrp}}$ (resp. $\hat{\mathbf{X}}_{\text{vgene}}$), we cluster the public dataset $\mathbf{Z}$ by averaging its CF-specific substitution profiles according to the `partis`-inferred [36] IMGT defined [39] V-subgroup (resp. V-gene) labels and then assign each row in $\mathbf{X}$ to a V-subgroup (resp. V-gene) cluster profile according to its V-subgroup (resp. V-gene) identity. We obtain the second set of profiles $\hat{\mathbf{X}}_{\text{naiveAA}}$ by using the `partis`-inferred naive sequences as substitution profiles (these profiles contain zeros and ones because they are based on one sequence only); we re-emphasize that these naive sequences are inferred based only on the corresponding subsampled sequences in $\mathbf{X}$. We cluster the public dataset $\mathbf{Z}$ once more by running K-means clustering based on the inferred naive sequences in $\mathbf{Z}$ and obtain our third set of substitution profiles $\hat{\mathbf{X}}_{\text{naiveAA}-\text{clust}}$ by assigning each CF in $\mathbf{X}$ to its closest cluster centroid. The additional cluster profiles $\hat{\mathbf{X}}_{\text{clust}}$ are obtained similarly as above, except in this case, we run K-means clustering based on the original frequency profiles in $\mathbf{Z}$. The K-means clustering procedure is run over a grid of cluster sizes ranging from 2 to 120 using the algorithm described by [40] with the standard euclidean distance metric. Lastly, the tensor $\hat{\mathbf{X}}_{\text{neut}}$ contains the simulated S5F neutral substitution profiles, which are described in the previous subsection.

The frequency tensors $\hat{\mathbf{X}}_{\text{vsubgrp}}$ and $\hat{\mathbf{X}}_{\text{vgene}}$ are important to include in our analysis because these profiles capture substitution information at the level of the V subgroup (V1, V2, . . .) and V gene (V1-5, V2-2, . . .), respectively; this is similar to the types of profiles obtained in [16]. Even though we expect the $\hat{\mathbf{X}}_{\text{vsubgrp}}$ and $\hat{\mathbf{X}}_{\text{vgene}}$ tensors to be correlated, we are interested in seeing whether either of these profiles will dominate the other in our regression model. As described in the introduction, most germinal center lineages do not accumulate many mutations relative to the naive sequence so substitution profiles based solely on the naive sequence (like $\hat{\mathbf{X}}_{\text{naiveAA}}$) may be informative for predicting the mutational patterns at conserved residue positions. In addition, we believe that the $\hat{\mathbf{X}}_{\text{naiveAA}-\text{clust}}$ cluster profiles are useful as the naive sequence can greatly influence the pattern of substitutions in a CF due to local sequence context. Unlike the $\hat{\mathbf{X}}_{\text{vsubgrp}}$ and $\hat{\mathbf{X}}_{\text{vgene}}$ substitution profiles, which are based on IMGT labeling schemes, the profiles in $\hat{\mathbf{X}}_{\text{naiveAA}-\text{clust}}$ (and $\hat{\mathbf{X}}_{\text{clust}}$) are determined by a data-driven clustering procedure, which allows us to group CFs in $\mathbf{Z}$ in a more intricate fashion. The simulated neutral substitution profiles $\hat{\mathbf{X}}_{\text{neut}}$ are able to provide some insight into the CF-specific SHM processes without the corresponding clonal selection effects.

To condense our model presentation, we introduce a four-dimensional tensor $\mathbf{X}^*$ that combines as many of the input profiles mentioned previously as we would like, where p, the size of the fourth tensor dimension, represents the number of external profiles used. We define $\mathbf{X}^* \equiv \{X^*_{i,j,k,l}\}$ to be the input data tensor that incorporates all the external information we want to use in our substitution profile predictions; note that $i \in \{1, \ldots, N_{CF}\}$ (N_{CF} CFs in the tensors), $j \in \{1, \ldots, 149\}$ (149 AHo positions), $k \in \{1, \ldots, 20\}$ (20 amino acids), and $l \in \{1, \ldots, p\}$ (p external profiles). Each element $X^*_{i,j,k,l}$ represents an amino acid frequency as described above for $\mathbf{T}_{i,j,k}$; for instance, $X^*_{5,130,1,4}$ represents the amino acid frequency of the first amino acid (i.e. alanine) at the 130th AHo position for the 5th CF in the 4th profile in the tensor. In addition, we use the indexing symbol $\bullet$ to extract all elements of a particular array dimension of a tensor (i.e. $\mathbf{X}^*_{10,50,\bullet,2}$ specifies the full substitution profile of the 20 amino acids at the 50th AHo position for the 10th CF in the 2nd profile in the tensor). This setup allows us to easily include as many external profiles as we would like.

Model formulation

Given the subsampled profiles $\mathbf{X}$ and all the external profiles $\mathbf{X}^*$, we compute a weighted average to form an estimator of $\mathbf{Y}$. Our independent-across-sites model $F(\mathbf{X}) = [f(\mathbf{X}_{\bullet,1,\bullet}), \ldots, f(\mathbf{X}_{\bullet,149,\bullet})]$ is specified as follows:

$$f(\mathbf{X}_{\bullet,j,\bullet}) \equiv f(\mathbf{X}_{\bullet,j,\bullet}; \boldsymbol{\alpha}_{j,\bullet}) = \sum_{l=1}^{p} \alpha_{j,l} \cdot \mathbf{X}^*_{\bullet,j,\bullet,l} + (1 - \sum_{l=1}^{p} \alpha_{j,l}) \cdot \mathbf{X}_{\bullet,j,\bullet}, \tag{1}$$

where $\boldsymbol{\alpha} = \{\alpha_{j,l}\}$; $0 \leq \alpha_{j,l} \leq 1$; $0 \leq \sum_{l=1}^{p} \alpha_{j,l} \leq 1$ represents the site-specific weights of the different external profiles for $j = 1, \ldots, 149$ and $l = 1, \ldots, p$. Although we consider f to be a function of the per-site data $\mathbf{X}_{\bullet,j,\bullet}$, the frequencies $\mathbf{X}^*_{\bullet,j,\bullet,l}$ are computed using sequence-level, site-dependent information. With $149 \times p$ parameter values of $\boldsymbol{\alpha}$, this is a highly parameterized model so we include regularization terms to prevent overfitting and obtain sparse, interpretable parameter estimates. Specifically, we use standard and spatial (fused) lasso penalties to achieve these goals.

Standard lasso penalties shrink individual parameters to zero and are commonly used to obtain sparse solutions in regression problems [23]. It has been shown that regression models using standard lasso penalties provide more accurate predictions than models using best subset

selection penalties when there is a low signal-to-noise ratio [41], which probably holds true in our problem as well. In addition, standard lasso penalties are convex functions, which is important in a regression problem as it guarantees that a local minimum is indeed a unique global solution [42].

On the other hand, fused lasso penalties shrink the differences between parameters to zero and are useful in regression problems with spatially-related covariates [24]. We believe that the $\boldsymbol{\alpha}$ parameters have a spatial relationship (i.e. adjacent residues are under similar constraints); for instance, given that the mutations in the framework regions are largely related to antibody stability, it makes sense that we would weight external profile information similarly in those regions. The fusion penalty in this setting enforces smoothness of the $\boldsymbol{\alpha}$ trend across the AHo positions. For example, if we penalize first-order differences of the $\boldsymbol{\alpha}$ trend, the fitting procedure will necessarily favor trends that have no slope (i.e. that are piecewise constant). We can obtain more flexible piecewise polynomial $\boldsymbol{\alpha}$ trends by penalizing higher-order successive differences of $\boldsymbol{\alpha}$ [25].

In our modeling framework, the standard lasso penalty is represented as $\sum_{j=1}^{149}\sum_{l=1}^{p}|\alpha_{j,l}| = \|\boldsymbol{\alpha}\|_1$ and the fused lasso penalty is specified by $\sum_{l=1}^{p}\|\nabla^d(\boldsymbol{\alpha}_{\bullet,l})\|_1$, where $\|\cdot\|_q$ denotes the L_q norm and $\nabla^d(\cdot)$ represents the dth difference operator. This $\nabla^d(\cdot)$ operator accepts a vector $\mathbf{v}$ as input (call its length $n_{\mathbf{v}}$) and outputs a length-$(n_{\mathbf{v}} - d)$ vector that results from successively differencing adjacent elements d times. In the special case when $d = 1$, the fusion penalty becomes $\sum_{l=1}^{p}\|\nabla^1(\boldsymbol{\alpha}_{\bullet,l})\|_1 = \sum_{j=2}^{149}\sum_{l=1}^{p}|\alpha_{j,l} - \alpha_{j-1,l}|$; the $|\alpha_{j,l} - \alpha_{j-1,l}|$ terms can be interpreted as first-order discrete derivatives.

Our unpenalized objective function can be written as:

$$L_2^{\boldsymbol{\alpha}} \equiv L_2^{\boldsymbol{\alpha}}(\mathbf{Y}, F(\mathbf{X})) = \frac{1}{149 \cdot N_{CF}}\sum_{j=1}^{149}||\mathbf{Y}_{\bullet,j,\bullet} - f(\mathbf{X}_{\bullet,j,\bullet}; \boldsymbol{\alpha}_{j,\bullet})||_2^2, \tag{2}$$

where, as in the last subsection, N_{CF} denotes the number of CFs in $\mathbf{X}$ and $\mathbf{Y}$; we refer to this objective as "L_2 Error". Our penalized estimation problem is defined in the following manner:

$$\begin{aligned}\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\ & L_2^{\boldsymbol{\alpha}}(\mathbf{Y}, F(\mathbf{X})) + \lambda_1\|\boldsymbol{\alpha}\|_1 + \lambda_2\sum_{l=1}^{p}\|\nabla^d(\boldsymbol{\alpha}_{\bullet,l})\|_1, \\ & \text{s.t. } 0 \le \alpha_{j,l} \le 1,\ \ 0 \le \sum_{l=1}^{p}\alpha_{j,l} \le 1,\ \ \forall j, l,\end{aligned} \tag{3}$$

where $\lambda_1, \lambda_2 \geq 0$ and $d \in \mathbb{N}$ signify tuning parameters. The differencing order d is used to specify a given level of smoothness in the spatial $\boldsymbol{\alpha}$ trend estimates because the $\sum_{l=1}^{p}\|\nabla^d(\boldsymbol{\alpha}_{\bullet,l})\|_1$ term in the above minimization problem encourages $\boldsymbol{\alpha}$ trends that have dth order discrete derivatives close to 0 (i.e. that are piecewise polynomials of order $d - 1$). In addition, careful selection of λ_1 and λ_2 is required to obtain an adequate model fit. Unfortunately, this is a constrained optimization problem with a multivariate output and there are not any obvious ways to minimize such an objective without resorting to general-purpose optimizers. Therefore, in all our experiments, we use the L-BFGS-B algorithm [43] to fit the above model. We note that the above penalized optimization problem is (non-strictly) convex so any local minimum is, in fact, a global solution too.

Jaccard similarity

While the model described above has computational and statistical appeal, in engineering applications it is mostly interesting to know the high-frequency amino acid predictions;

however, our penalized objective function focuses attention on the complete substitution profiles and not exclusively the high-frequency amino acids. To provide a metric with exclusive focus on high-frequency amino acids, we utilize the Jaccard similarity metric, which can be used to measure differences between predicted and observed sets. Sets of high-frequency amino acids are defined at each position by a minimum frequency cutoff t; Jaccard similarities are then computed between the observed and predicted sets and averaged across each CF and AHo position in the dataset.

The Jaccard similarity metric [22] measures the similarity between two finite sets. Specifically, for any sets A and B, the similarity metric $J(A, B)$ is defined as the ratio of the intersection size $|A \cap B|$ to the union size $|A \cup B|$. It has these properties: $0 \leq J(A, B) \leq 1$; $J(A, B) = 1$ when $A = B$ and $J(A, B) = 0$ when $A \cap B = \emptyset$ (empty set). To formally establish our use of Jaccard similarity, we define the following notation. Let $\mathcal{Y}_{i,j} = \{y \in \mathbf{Y}_{i,j,\bullet} \mid y \geq t\}$ represent the set of amino acid frequencies at AHo position j for CF i that has observed frequencies greater than or equal to the cutoff t and denote $\boldsymbol{\mathcal{Y}} \equiv \{\mathcal{Y}_{i,j}\}$ for $i = 1, \ldots, N_{CF}$ and $j = 1, \ldots, 149$. We define $\hat{\boldsymbol{\mathcal{F}}}^{\mathbf{X}}_{i,j}$ and $\hat{\mathcal{F}}^{\mathbf{X}} \equiv \{\hat{\mathcal{F}}^{\mathbf{X}}_{i,j}\}$ to be the analogous quantities for the predicted amino acid frequencies. If we let $\mathcal{A}(\mathcal{Y}')$ denote a function that accepts as input an amino acid frequency set $\mathcal{Y}'$ (i.e. $\mathcal{Y}_{i,j}$ or $\hat{\mathcal{F}}^{\mathbf{X}}_{i,j}$) and outputs the corresponding set of amino acid identities, then our Jaccard similarity objective can be written as:

$$J^{\boldsymbol{\alpha}}_t \equiv J^{\boldsymbol{\alpha}}_t(\mathbf{Y}, F(\mathbf{X})) = \frac{1}{149 \cdot N_{CF}} \sum_{i=1}^{N_{CF}} \sum_{j=1}^{149} J\left(\mathcal{A}(\mathcal{Y}_{i,j}), \mathcal{A}(\hat{\mathcal{F}}^{\mathbf{X}}_{i,j})\right), \tag{4}$$

which is referred to as the "Jaccard Similarity" objective. We can define a penalized Jaccard estimation problem by substituting $-J^{\boldsymbol{\alpha}}_t(\mathbf{Y}, F(\mathbf{X}))$ for $L^{\boldsymbol{\alpha}}_2(\mathbf{Y}, F(\mathbf{X}))$ in Eq (3). Jaccard similarity optimization is difficult using derivative-based optimization because of its discrete nature, so we use a smooth approximation of the aforementioned metric for model fitting in our experiments (see S2 Text for detailed explanation).

Forward stepwise selection

We devise a forward stepwise selection procedure to help us determine the combination of external profiles that best predict the outcome of interest, which can be penalized L_2 Error or Jaccard Similarity. In this procedure, we initially try all possible external profiles in the model separately and determine the best fit using 5-fold cross-validation. We cache the best model from the initial step and continue fitting models with two external profiles; the first external profile is fixed to be the best profile from the previous round and the second profile can be any possible remaining external profile. We continue this iterative scheme until we reach a prespecified limit on the number of external profiles allowed in $\mathbf{X}^*$. It is important to note that to ease computation, we perform forward selection using the unpenalized variants of our models. Even though this procedure is greedy and not as thorough as all-subsets selection, we believe this technique provides the best trade-off between accuracy and efficiency. We provide the implementation of our stepwise procedures at https://github.com/krdav/SPURF.

Inference pipeline

We apply a 80%/20% training/test split to the model fitting dataset described above. We first run the forward stepwise selection procedure with a maximum profile limit of five to approximately determine the best profile groupings starting with a single profile and ending with a group of five profiles. Using the profile groupings from the previous step, we fit the penalized

version of the model and use 5-fold cross-validation to obtain estimates of the relevant tuning parameters, which consist of the lasso penalty weights λ_1, λ_2 and the differencing order d; note that we report unpenalized performance estimates when we run cross-validation. After we determine the optimal tuning parameters via cross-validation, we fit the penalized model using the entire training portion of the model fitting dataset and the best tuning parameters and cache the resulting parameter estimates of $\boldsymbol{\alpha}$. Once we obtain the estimates of $\boldsymbol{\alpha}$ from the penalized model, we can use them to compute the chosen performance metric on the testing portion of the model fitting dataset and any other validation dataset of interest.

Results

As described in the methods (the Inference Pipeline subsection), we first need to infer the best profile groupings to use in penalized model fitting. To determine these groupings, we run the forward stepwise selection procedure for both the L_2 error function and the smoothed Jaccard objective function with a frequency cutoff $t = 0.2$ (Table 2).

For both objective functions, the forward selection path is the same until $\mathbf{X}^* = \{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}, \hat{\mathbf{X}}_{\text{neut}}, \hat{\mathbf{X}}_{\text{vsubgrp}}\}$. For the L_2 loss function, model performance is the best when $\mathbf{X}^* = \{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}, \hat{\mathbf{X}}_{\text{neut}}, \hat{\mathbf{X}}_{\text{vsubgrp}}\}$ even though there are diminishing returns for using profiles beyond $\mathbf{X}^* = \{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}\}$. In a similar fashion, the Jaccard similarity estimates tend to be highest when $\mathbf{X}^* = \{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}\}$, despite the almost identical model performance from just using $\mathbf{X}^* = \{\hat{\mathbf{X}}_{\text{naiveAA}}\}$. For the subsequent penalized model fitting step, we choose to evaluate the $\{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}, \hat{\mathbf{X}}_{\text{neut}}\}$ and $\{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}, \hat{\mathbf{X}}_{\text{neut}}, \hat{\mathbf{X}}_{\text{vsubgrp}}\}$ profile groupings with the L_2 objective and $\{\hat{\mathbf{X}}_{\text{naiveAA}}\}$ and $\{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}\}$ with the smoothed Jaccard similarity objective. The inclusion of the $\hat{\mathbf{X}}_{\text{vgene}}$ tensor puts a notable restriction on the model; no prediction can be made for a sequence annotated to a V gene which has not been observed in our Public dataset.

We now use the approximate profile groupings obtained from the forward stepwise selection procedure to fit our regularized models. The penalized estimation problem has additional tuning parameters that must be determined. In our experiments, we cross-validate over penalty parameters; $\lambda_1, \lambda_2 = 10^{-7}, 5.05 \times 10^{-6}, 10^{-5}$; the differencing order, $d = 1, 2, 3$; and the two profile groupings specified above for both the L_2 error and Jaccard similarity objectives. The best regularized L_2 model uses $\mathbf{X}^* = \{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}, \hat{\mathbf{X}}_{\text{neut}}, \hat{\mathbf{X}}_{\text{vsubgrp}}\}$, while the best regularized Jaccard model utilizes $\mathbf{X}^* = \{\hat{\mathbf{X}}_{\text{naiveAA}}\}$ (S1 Table and S7 Fig). In summary, using many external profiles is important for predicting the complete substitution profiles, while the inferred

Table 2. Results of forward stepwise selection on our L_2 and smooth Jaccard objective functions.

Objective Function	Unregularized CV					
L_2 Error	∅	$\hat{\mathbf{X}}_{\text{naiveAA}}$	$\hat{\mathbf{X}}_{\text{vgene}}$	$\hat{\mathbf{X}}_{\text{neut}}$	$\hat{\mathbf{X}}_{\text{vsubgrp}}$	$\hat{\mathbf{X}}_{\text{naiveAA-clust-5}}$
	0.110	0.0542	0.0459	0.0456	0.0455	0.0456
Jaccard Similarity ($t = 0.2$)	∅	$\hat{\mathbf{X}}_{\text{naiveAA}}$	$\hat{\mathbf{X}}_{\text{vgene}}$	$\hat{\mathbf{X}}_{\text{neut}}$	$\hat{\mathbf{X}}_{\text{vsubgrp}}$	$\hat{\mathbf{X}}_{\text{naiveAA-clust-85}}$
	0.9170	0.9322	0.9324	0.9323	0.9319	0.9318

The performance estimates shown in the table are obtained using 5-fold cross-validation. Going from left to right, each column represents the best profile addition into $\mathbf{X}^*$ with the associated CV performance estimate. For Jaccard, we fit using the smooth Jaccard objective, but report exact Jaccard similarity estimates, both using frequency cutoff $t = 0.2$. Note that we fix the prespecified limit on the number of external profiles allowed in $\mathbf{X}^*$ to be 5. ∅ represents the model using only the input sequence.

https://doi.org/10.1371/journal.pcbi.1006388.t002

naive sequence is the only external profile deemed useful for our model to accurately predict the observed high-frequency amino acids (where high-frequency is defined by being at least 20% of the observed amino acids).

Our optimization times for both L_2 loss and Jaccard Similarity on the 500 CFs ranged from 12 to 15 minutes. Our optimization is based on evaluating the objective function at different points and each objective function call has linear complexity in the number of CFs so increasing the number of CFs should result, on average, in a linear increase in time complexity. Computational time invested in pre-processing is one-time and negligible.

In addition to predictive performance, we are also interested in understanding how the estimated parameter weights from our best regularized L_2 model vary across the different external profiles in $\mathbf{X}^*$ and antibody regions. For convenience, we aggregate the estimates of $\boldsymbol{\alpha}$ associated with the V gene ($\hat{\mathbf{X}}_{\text{vgene}}$ and $\hat{\mathbf{X}}_{\text{vsubgrp}}$) and with the full naive sequence ($\hat{\mathbf{X}}_{\text{naiveAA}}$ and $\hat{\mathbf{X}}_{\text{neut}}$) as these sets of profiles are intuitively similar (Fig 3); the V-gene and V-subgroup profiles are both derived by averaging over different IMGT V germline gene labeling schemes and the simulated S5F neutral substitution profiles originate from the CF-specific inferred naive sequence. Antibody heavy chain (and light chain) sequences can be partitioned into framework regions (FWKs) and complementarity-determining regions (CDRs) by the AHo definitions [21]; the BCR binding affinity is largely determined by the CDRs (especially by the heavy chain CDR3), while the FWKs encode the structural constraints of the BCR and thus can be strongly conserved [44]. The $\hat{\mathbf{X}}_{\text{vgene}}$ and $\hat{\mathbf{X}}_{\text{vsubgrp}}$ profiles are extremely important for prediction at FWK1-FWK3, which is not surprising as V germline genes extend from the FWK1 to the beginning of the CDR3. In contrast, the $\hat{\mathbf{X}}_{\text{naiveAA}}$ and $\hat{\mathbf{X}}_{\text{neut}}$ external profiles are heavily weighted in the CDR3 and FWK4; this result is also intuitive because the CDR3 is highly variable across CFs as it is a strong determinant of antigen-binding specificity, the $\hat{\mathbf{X}}_{\text{naiveAA}}$ and $\hat{\mathbf{X}}_{\text{neut}}$ profiles are our only CF-specific sources of external information, and the V gene specific profiles cannot provide any information beyond the end of the V gene. Furthermore, the FWKs have, on average, more support from the external profiles compared to the CDRs, which is consistent with our understanding of antibody biochemistry as the FWKs are structurally constrained and thus need to be more conserved compared to the more flexible CDRs. We note that the middle of the CDR3 has artificially low estimates of $\boldsymbol{\alpha}$ because most of the AHo positions in the CDR3 have only a few or no defined sequence positions in the dataset (S1 Fig).

While our penalized modeling framework allows for easy interpretation of the parameter estimates, ultimately the quality of the $\boldsymbol{\alpha}$ estimates is determined by their performance on independent test datasets. Specifically, we compute the L_2 error ($L_2^{\boldsymbol{\alpha}}$) and Jaccard similarity ($J_{0.2}^{\boldsymbol{\alpha}}$) between the predicted and observed profiles associated with both the testing portion of the model fitting dataset and the Briggs validation dataset (Table 3); we remind readers that

Table 3. The model performance using either L_2 error or Jaccard similarity resulting from predicting on independent datasets.

Objective Function	Model Type	Objective Function Values		
		Model fitting: test	Briggs	Liao
L_2 Error	Best	0.0492	0.0511	0.0991
	Baseline	0.114	0.129	0.183
Jaccard Similarity ($t = 0.2$)	Best	0.9289	0.9227	0.8516
	Baseline	0.9156	0.9053	0.8439

We provide results for the testing portion of the model fitting dataset, the Briggs validation dataset, and the Liao dataset. Note that the term "baseline" refers to predictions made using only the input sequence (i.e. model predictions with all parameter values of $\boldsymbol{\alpha}$ set to 0). Lower L_2 error and higher Jaccard Similarity mean higher accuracy.

https://doi.org/10.1371/journal.pcbi.1006388.t003

Table 4. Mode prediction results from both the testing portion of the model fitting dataset and the Briggs dataset fitted using the L_2 objective function. For each CF and AHo position in a given dataset, we determine whether the predicted mode (i.e. highest-frequency amino acid) from our best model is the same as the actual mode. Results are aggregated based on whether or not the input sequence has the correct mode. At the left side of the vertical bar (|) is the count for the germline predicted modes (i.e. situations when the predicted amino acid mode is the naive sequence amino acid) and at the right side is the count for the non-germline predicted modes (vice-versa).

germline | non-germline		Correct SPURF Mode Prediction?	
		Yes	No
Is input amino acid the mode?	Yes	10,473 | 465	156 | 0
	No	349 | 0	170 | 395

(**a**) Model fitting: test dataset

germline | non-germline		Correct SPURF Mode Prediction?	
		Yes	No
Is input amino acid the mode?	Yes	10,541 | 376	178 | 1
	No	474 | 0	196 | 393

(**b**) Briggs dataset

https://doi.org/10.1371/journal.pcbi.1006388.t004

these predictions are made based on the subsampled (i.e. single-sequence) profiles in the aforementioned datasets and compared to the corresponding actual amino acid frequencies through the L_2^{α} and $J_{0.2}^{\alpha}$ performance metrics (Fig 2). Additionally, we compute the L_2 error and Jaccard similarity on all sequences in the Liao dataset, comparing the baseline and SPURF predictions to the full amino acid frequencies (Table 3, S2 Table, and S6 Fig). Our model improves upon the "baseline" prediction performance, where "baseline" refers to predictions made using only the input sequence (i.e. model predictions with all parameter values of $\boldsymbol{\alpha}$ set to 0).

In addition, we also want to know how well our model performs in the different antibody regions (i.e. FWKs/CDRs). To answer this question, we compute the same metrics as shown in Table 3 for the different FWKs and CDRs (Fig 4). To provide some insight into the variability

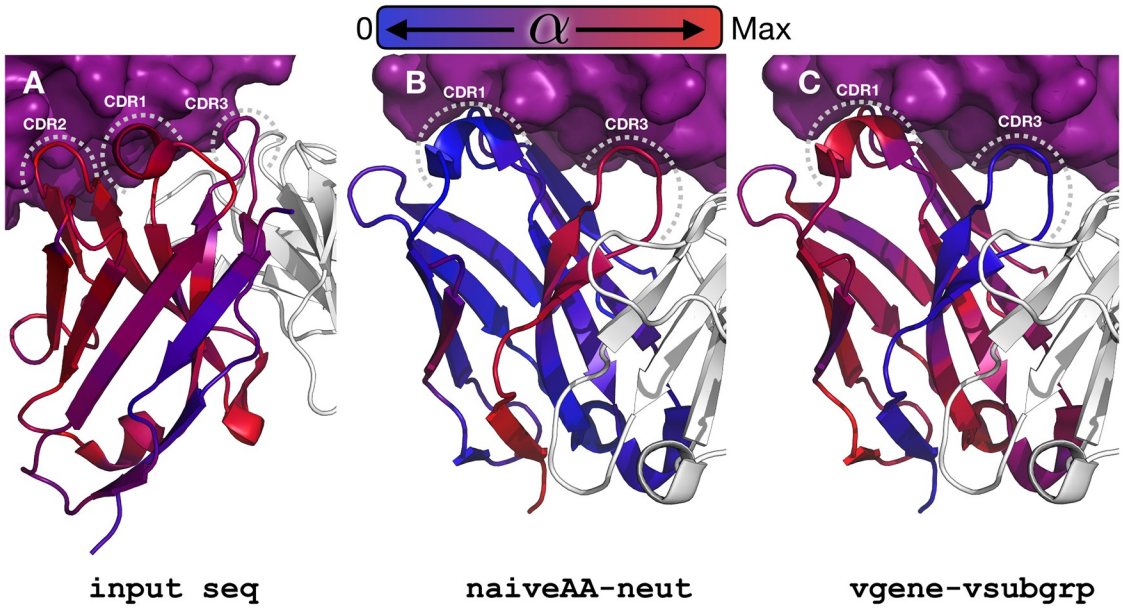

Fig 5. Positional profile weights α mapped to an antibody protein structure (PDB: 5X8L). The antigen (PD-L1) appears as a purple surface at the top of the images, the light chain appears in white cartoon, and the heavy chain is displayed using a blue to red color gradient; the grey dashed lines mark the CDR loops. The color gradient represents the possible values of profile weights in $\boldsymbol{\alpha}$ and goes from blue at a zero weight to red at the maximum weight for the profile. The display in panels **B** and **C** is rotated relative to panel **A** to better show results for CDR1 and CDR3; as a consequence, the CDR2 loop is hidden behind the CDR1. Panel **A** shows that the input sequence has high weight at the CDR1 and CDR2, panel **B** illustrates that the naive sequence and the neutral substitution profile have high weight at the CDR3 and FWK4, and panel **C** demonstrates that the V gene and V subgroup profiles are highly weighted in parts of the CDR1 but more generally in the FWKs, especially at the heavy and light chain interface.

https://doi.org/10.1371/journal.pcbi.1006388.g005

of the model performance estimates in the different regions, we calculate bootstrap standard errors, which are expressed as error bars in Fig 4.

We see that our substitution profile prediction model performs well in the CDRs relative to the baseline model. This is an important finding because antigen binding is largely determined by the sequence segments in the CDRs, and especially CDR3. In fact, our models seem to provide the greatest improvement in performance in the CDR3, which is also the hardest region to predict because it has the highest amount of sequence variability. Another important takeaway is that the prediction performance is better in FWKs than CDRs, which is presumably because FWKs have lower variance and are more conserved compared to CDRs. In summary, our prediction models are able to systematically integrate different data sources to make better predictions of the per-site amino acid compositions in CFs.

Our model also improves the prediction of the highest-frequency amino acid at a given position, referred to here as the mode (Table 4). Indeed, the counts in the bottom-left cells (cases where the model is correctly predicting the actual mode given an incorrect input sequence amino acid) are larger than the counts in the top-right cells (vice-versa). In addition, the input sequence amino acids that are not the true modes but correctly predicted by the model to be the actual modes are all germline reversions, which is consistent with the $\hat{\mathbf{X}}_{\text{naiveAA}}$ profile being heavily weighted in our prediction model (Fig 3). In the opposite case, where the input sequence amino acid is correct but the model prediction is wrong, all the counts consist of germline predictions as well. In summary, many of the mode predictions are just germline reversions and, in fact, most of these predictions are to the true modes (i.e. the actual highest-frequency amino acids); however, most of the input sequence amino acids are the true modes already ($\approx$ 99%).

The in-sample and out-of-sample prediction performances demonstrate that our SPURF inference pipeline is able to obtain accurate and robust estimates of $\boldsymbol{\alpha}$. Specifically, prediction performance is consistently similar but slightly worse when comparing the Briggs dataset to the model fitting test set, which likely reflects two things: 1) the median number of sequences per CF in the Briggs set is lower than in the test set (Tables 1 and 2) the model fitting dataset is sampled from the same donors as the dataset for cross-validation. Regardless, the differences between the test and Briggs datasets are small, which provides evidence in support of our model performance estimates. We also note that the test on the Liao data yielded results strongly favoring SPURF over baseline. Since the Liao dataset carries a high mutation frequency compared to the average CF of the other dataset it is (as expected) harder to predict the amino acid frequency, which is reflected in the magnitude of both the L_2 error and Jaccard similarity for all predictions. Subjective assessments of the inferred substitution profiles coincide with our description of the L_2 error metric, namely that fine-grained amino acid substitution information is captured by SPURF (S2 Fig).

The SPURF model setup produces interpretable and meaningful profile weights (Fig 5; per-profile decomposition in S3 Fig). The input sequence is strongly weighted in the CDRs, indicating that substitutions in these regions are both specific and conserved within the CF and, therefore, cannot easily utilize the information from other CFs. The weight on the V gene specific profiles spikes at CDR1 and at the end of FWK3, which is at the heavy and light chain interface. We note that, as expected, the weight on the V gene specific profiles is minimal downstream of FWK3 as this is the end of the V gene and the beginning of the V-D junction region. As such, nothing prevents the V gene profiles from having a high weight downstream of FWK3, but the model framework has chosen these meaningful weights without any manual interference. We ascribe this shrinkage feature of the weights to the standard lasso penalty built into SPURF. The profiles that are derived from the inferred naive sequence ($\hat{\mathbf{X}}_{\text{naiveAA}}$,

$\hat{\mathbf{X}}_{\text{neut}}$) take up the missing weight of the V gene profiles as these are highly weighted in the CDR3 and FWK4.

Discussion

In this paper, we present SPURF, a statistical framework for predicting CF-specific amino acid frequency profiles from single input BCR sequences by leveraging multiple sources of external information. We use standard and spatial lasso penalties to prevent our model from overfitting and obtain sparse, interpretable estimates of the profile weights, expressed by an $\boldsymbol{\alpha}$ matrix. The spatial lasso penalizes extreme differences between spatially-adjacent profile weights, while the standard lasso penalties promote simpler models by shrinking parameter values in $\boldsymbol{\alpha}$ to 0 if the associated external profiles are not useful predictors. We show that our method not only performs well on the held-out (test) portion of our model fitting dataset but also provides accurate predictions on the Briggs and Liao external validation datasets. Indeed, we did not obtain the Briggs or Liao validation datasets until after we ran our model inference pipeline on the model fitting dataset.

Using two different objective functions we fitted SPURF to predict the frequencies of all amino acids (L_2 objective) and only the >20% frequency amino acids (Jaccard similarity objective). With the L_2 objective we obtained a large difference (0.114 to 0.0492) between the baseline model and SPURF, which was confirmed using repertoire wide data from [12] and single clone data from [38] (Fig 4 and S6 Fig). With the Jaccard similarity objective improvements over baseline were more modest (0.9156 to 0.9289) showing that SPURF is strongest at predicting the full spectrum of amino acid frequencies (Table 3). Still, fitted using the L_2 objective, SPURF can recover the highest frequency amino acid of a clonal family (mode prediction) much better than a random sequence from the corresponding clonal family (Table 4), showing the versatility of the L_2 objective.

Our work can be seen as a prediction-based extension of the work of [16] and [19]. This previous work illustrates that amino acid substitution profiles differ between germline genes, a finding supported by the context specificity of somatic hypermutation [20]. In our work, we provide a prediction algorithm that takes a single BCR sequence from a clonal family as input and outputs a CF-specific substitution profile estimate for the whole VDJ region. As SPURF relies on large CFs to establish a ground truth substitution profile it is possible that certain types of rare clones or V/J gene combinations are not included in our training/test data. For such rare events the error estimates reported cannot be reliably used, however, we note that our training/test data cover a broad set of V/J combinations (S4 Fig) and that the substitution profile of a rare, but expanded, broadly HIV neutralizing clone is well predicted (S6 Fig).

We believe that this work will be a useful tool for antibody engineering in situations when it is important to maintain antibody binding affinity to the same epitope. The predicted profiles from SPURF can be used to choose the sites that are most tolerable for mutagenesis and the substitutions that are most likely to maintain binding specificity; as such, this information can be used to engineer antibodies with better biophysical properties.

The seven datasets utilized in the present study were all derived from different laboratories employing varying strategies to obtain their processed data which served as input for SPURF. We carefully examined available resources and selected the datasets to be used in our model. However, our approach would greatly benefit from a large and uniformly accessible repository of Rep-seq datasets. For this to happen, data has to be discoverable and usable, including having all information about the study and data processing available together with the raw and processed data in publicly accessible data repositories. Recently, the Adaptive Immune Receptor Repertoire (AIRR) community [45] proposed MiAIRR [46], a set of minimal standard

elements to be published alongside the raw and processed data. Future Rep-seq studies following this initiative and making their data available under the MiAIRR-standard will facilitate the development of SPURF and future approaches with similar goals.

To our knowledge, SPURF is the first prediction algorithm for B cell CF substitution profiles. There are many possible extensions; in our SPURF inference pipeline, we subsample single BCR sequences from CFs to use as model input; unfortunately, this means that our modeling analysis is conditional on a dataset that does not account for the variability associated with the subsampling process. One obvious means of fixing the above problem is to draw multiple subsamples from each CF and treat these multiple "observations" per CF within a dataset as a clustered data or weighted least squares problem. In addition, our model fitting dataset consists of only the largest CFs because we need accurate CF-specific substitution profile estimates to serve as the ground truth. This non-random sampling technique could potentially bias our analysis results; however, this appears unlikely given our model's performance on the external Briggs and Liao validation datasets. Furthermore, our approach models per-site amino acid composition in a CF and accounts for interactions between sites only through the fusion lasso penalties. It is well known from other protein studies that spatially-adjacent amino acid residues evolve jointly [47, 48], presumably to maintain structural stability, or in the case of antibodies to stabilize the interface between heavy and light chains [49]. In the context of antibodies, residues in the FWKs have the potential to co-evolve (e.g. FWK residues flanking the CDRs could co-evolve to stabilize the stem leading to the more flexible CDRs). Thus, figuring out how to incorporate more detailed interaction effects in our model is an important avenue for future research.

Supporting information

S1 Text. Model interpretation. Details of the penalized regression model.
(PDF)

S2 Text. Smoothed Jaccard similarity. Explanation of the continuous approximation of the Jaccard similarity used in the optimization.
(PDF)

S3 Text. Supporting information references. References used in the Supporting Information sections.
(PDF)

S1 Table. The results from fitting the regularized models using 5-fold cross-validation. We present the optimal tuning parameters selected from $\lambda_1, \lambda_2 = 10^{-7}, 5.05 \times 10^{-6}, 10^{-5}$ and $d = 1, 2, 3$ and show the associated cross-validated performance estimates. Note that the possible choices of $\mathbf{X}^*$ for the L_2 error metric include the $\{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}, \hat{\mathbf{X}}_{\text{neut}}\}$ and $\{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}, \hat{\mathbf{X}}_{\text{neut}}, \hat{\mathbf{X}}_{\text{vsubgrp}}\}$ groupings, while the $\{\hat{\mathbf{X}}_{\text{naiveAA}}\}$ and $\{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}\}$ groupings are the possible $\mathbf{X}^*$ choices for the smoothed Jaccard similarity objective.
(TIFF)

S2 Table. The results from fitting the regularized models using 5-fold cross-validation. The unregularized and regularized model performance using either L_2 Error or Jaccard Similarity resulting from predicting on independent datasets. We provide results for the testing portion of the model fitting dataset, the Briggs validation dataset, and the Liao dataset. Note that the term "baseline" refers to predictions made using only the input sequence (i.e. model predictions with all parameter values of $\boldsymbol{\alpha}$ set to 0) and lower L_2 error and higher Jaccard Similarity is preferred.
(TIFF)

S1 Fig. A stacked barplot of the estimated parameter values of α from the best regularized L_2 model. The black vertical lines represent the boundaries between the different CDRs and FWKs. Due to the AHo antibody numbering used [21], some positions are assigned to a gap character (an AHo position that does not map to a sequence position). The percentage of CFs that are not assigned to gap characters is shown in the bottom plot for each AHo position. The input sequence is heavily weighted in regions with high gap percentages because of the standard lasso penalty included in our model. The conserved Tryptophan amino acid is observed as a spike in the $\hat{\mathbf{X}}_{\text{vgene}}$ and $\hat{\mathbf{X}}_{\text{naiveAA}}$ profile weights following the end of CDR1 (position 43 in the AHo scheme). The conserved Cysteine amino acid that defines the beginning of CDR3 is not readily observed, presumably because this is invariant in all profiles. Generally, the input sequence has less weight in CDR3 and FWK4, which indicates that there is some conservation during affinity maturation. Beyond CDR3 and FWK4, there is a general trend that the input sequence has higher weight in the CDRs than in the FWKs, which suggests that there is a higher level of conservation in the FWKs than in the CDRs during affinity maturation. A more surprising observation is the spike in the $\hat{\mathbf{X}}_{\text{vgene}}$, $\hat{\mathbf{X}}_{\text{vsubgrp}}$, and $\hat{\mathbf{X}}_{\text{neut}}$ weights at AHo position 83 near the beginning of FWK3 (the "outer" loop); this could indicate a conserved position not previously described.
(TIFF)

S2 Fig. A logo plot displaying the input sequence, predicted profile, and true profile (ordered from top to bottom) for an arbitrary CF in the Briggs dataset. The logos are plotted using AHo numbers (1-149) and AHo positions undefined in the sequence are shown as empty columns. The predicted profile (middle) captures much of the amino acid composition information associated with the full profile (bottom).
(TIFF)

S3 Fig. Positional profile weights α mapped to an antibody protein structure (PDB: 5X8L). The antigen (PD-L1) appears as a purple surface at the top of the images, the light chain appears in yellow cartoon, and the heavy chain is displayed using a blue to red color gradient. The color gradient represents the possible values of profile weights in $\boldsymbol{\alpha}$ and goes from blue at a zero weight to red at the maximum weight for the profile. The black dashed lines mark the CDR loops; note that the CDR2 loop is hidden behind the CDR1. The colored balls represent the AHo-defined FWK/CDR boundaries. The black arrows indicate regions of high profile weight. The $\hat{\mathbf{X}}_{\text{naiveAA}}$ profile is heavily weighted in CDR3 and FWK4. The $\hat{\mathbf{X}}_{\text{vgene}}$ profile weighting is fairly even from FWK1 through FWK3; it spikes slightly in CDR1 and completely disappears beyond FWK3, which is expected as the V-D junction region starts past the end of FWK3. The $\hat{\mathbf{X}}_{\text{neut}}$ profile weighting is fairly even across sites but spikes near the beginning of FWK3 (the "outer" loop). The $\hat{\mathbf{X}}_{\text{vsubgrp}}$ profile weighting is distributed similarly to that of the $\hat{\mathbf{X}}_{\text{vgene}}$ profile with the exception of a spike at the end of FWK3 (i.e. at the heavy and light chain interface).
(TIFF)

S4 Fig. Distribution of per-clonal-family V/J gene combination usage in the different dataset partitions. Minimum frequency of 1% in either partition used as a cutoff for inclusion.
(TIFF)

S5 Fig. Distribution of per-clonal-family V/J subgroup combination usage in the different dataset partitions. Minimum frequency of 1% in either partition used as a cutoff for inclusion.
(TIFF)

S6 Fig. Two histograms showing L_2 loss estimates based on model predictions from the baseline model and best model for the Liao dataset [38]. In this analysis, we made model predictions using each of the 312 sequences as the input sequence for our model.
(TIFF)

S7 Fig. A heatmap showing 5-fold cross-validated L_2 loss results from fitting the regularized models for the $\{\hat{\mathbf{X}}_{\text{naiveAA}}, \hat{\mathbf{X}}_{\text{vgene}}, \hat{\mathbf{X}}_{\text{neut}}, \hat{\mathbf{X}}_{\text{vsubgrp}}\}$ profile grouping. We present unregularized L_2 loss estimates for tuning parameters λ_1, $\lambda_2 = 10^{-7}, 5.05 \times 10^{-6}, 10^{-5}$ and the order of differencing, $d = 1, 2, 3$, and mark the optimal tuning parameters found from our experiments.
(TIFF)

S8 Fig. A plot of the function $f_\epsilon(a_i, 0.2)$ against $a_i \in [0, 1]$ for various values of ϵ. As ϵ gets larger, $f_\epsilon(a_i, 0.2)$ tends to the indicator function $f(a_i, 0.2)$.
(TIFF)

Acknowledgments

We would like to thank Jason A. Vander Heiden and Steven H. Kleinstein for sharing post-processed data (dataset 1-4), Mikhail Shugay for sharing post-processed data (dataset 5), Uri Hershberg for providing the ImmuneDB data (dataset 6). We would also like to thank Juno Therapeutics, Inc. for providing and preparing the single cell dataset used as our external validation.

Author Contributions

Conceptualization: Kristian Davidsen.

Data curation: Kristian Davidsen.

Formal analysis: Amrit Dhar, Kristian Davidsen.

Methodology: Amrit Dhar, Kristian Davidsen, Frederick A. Matsen, IV, Vladimir N. Minin.

Project administration: Frederick A. Matsen, IV, Vladimir N. Minin.

Software: Amrit Dhar, Kristian Davidsen.

Supervision: Frederick A. Matsen, IV, Vladimir N. Minin.

Validation: Amrit Dhar, Kristian Davidsen.

Visualization: Amrit Dhar, Kristian Davidsen.

Writing – original draft: Amrit Dhar, Kristian Davidsen.

Writing – review & editing: Amrit Dhar, Kristian Davidsen, Frederick A. Matsen, IV, Vladimir N. Minin.

References

1. Igawa T, Tsunoda H, Kuramochi T, Sampei Z, Ishii S, Hattori K. Engineering the variable region of therapeutic IgG antibodies. mAbs. 2011; 3(3):243–252. https://doi.org/10.4161/mabs.3.3.15234 PMID: 21406966
2. Clark RH, Latypov RF, De Imus C, Carter J, Wilson Z, Manchulenko K, et al. Remediating agitation-induced antibody aggregation by eradicating exposed hydrophobic motifs. mAbs. 2014; 6(6): 1540–1550. https://doi.org/10.4161/mabs.36252 PMID: 25484048
3. Casaz P, Boucher E, Wollacott R, Pierce BG, Rivera R, Sedic M, et al. Resolving self-association of a therapeutic antibody by formulation optimization and molecular approaches. mAbs. 2014; 6(6): 1533–1539. https://doi.org/10.4161/19420862.2014.975658 PMID: 25484044

4. Courtois F, Agrawal NJ, Lauer TM, Trout BL. Rational design of therapeutic mAbs against aggregation through protein engineering and incorporation of glycosylation motifs applied to bevacizumab. mAbs. 2016; 8(1):99–112. https://doi.org/10.1080/19420862.2015.1112477 PMID: 26514585

5. Geoghegan JC, Fleming R, Damschroder M, Bishop SM, Sathish HA, Esfandiary R. Mitigation of reversible self-association and viscosity in a human IgG1 monoclonal antibody by rational, structure-guided Fv engineering. mAbs. 2016; 8(5):941–950. https://doi.org/10.1080/19420862.2016.1171444 PMID: 27050875

6. Harding FA, Stickler MM, Razo J, DuBridge R. The immunogenicity of humanized and fully human antibodies: residual immunogenicity resides in the CDR regions. mAbs. 2010; 2(3):256–265. https://doi.org/10.4161/mabs.2.3.11641 PMID: 20400861

7. McConnell AD, Zhang X, Macomber JL, Chau B, Sheffer JC, Rahmanian S, et al. A general approach to antibody thermostabilization. mAbs. 2014; 6(5):1274–1282. https://doi.org/10.4161/mabs.29680 PMID: 25517312

8. Seeliger D, Schulz P, Litzenburger T, Spitz J, Hoerer S, Blech M, et al. Boosting antibody developability through rational sequence optimization. mAbs. 2015; 7(3):505–515. https://doi.org/10.1080/19420862.2015.1017695 PMID: 25759214

9. Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences. 1992; 89(22):10915–10919. https://doi.org/10.1073/pnas.89.22.10915

10. Tas JM, Mesin L, Pasqual G, Targ S, Jacobsen JT, Mano YM, et al. Visualizing antibody affinity maturation in germinal centers. Science. 2016; 351(6277):1048–1054. https://doi.org/10.1126/science.aad3439 PMID: 26912368

11. Victora GD, Nussenzweig MC. Germinal Centers. Annual Review of Immunology. 2012; 30(1):429–457. https://doi.org/10.1146/annurev-immunol-020711-075032 PMID: 22224772

12. Briggs AW, Goldfless SJ, Timberlake S, Belmont BJ, Clouser CR, Koppstein D, et al. Tumor-infiltrating immune repertoires captured by single-cell barcoding in emulsion. bioRxiv. 2017; p. 134841.

13. Kepler TB, Munshaw S, Wiehe K, Zhang R, Yu JS, Woods CW, et al. Reconstructing a B-Cell Clonal Lineage. II. Mutation, Selection, and Affinity Maturation. Frontiers in Immunology. 2014; 5:170. https://doi.org/10.3389/fimmu.2014.00170 PMID: 24795717

14. Kuraoka M, Schmidt AG, Nojima T, Feng F, Watanabe A, Kitamura D, et al. Complex antigens drive permissive clonal selection in germinal centers. Immunity. 2016; 44(3):542–552. https://doi.org/10.1016/j.immuni.2016.02.010 PMID: 26948373

15. Schmidt AG, Xu H, Khan AR, O'Donnell T, Khurana S, King LR, et al. Preconfiguration of the antigen-binding site during affinity maturation of a broadly neutralizing influenza virus antibody. Proceedings of the National Academy of Sciences. 2013; 110(1):264–269. https://doi.org/10.1073/pnas.1218256109

16. Sheng Z, Schramm CA, Kong R, Mullikin JC, Mascola JR, Kwong PD, et al. Gene-specific substitution profiles describe the types and frequencies of amino acid changes during antibody somatic hypermutation. Frontiers in Immunology. 2017; 8:537. https://doi.org/10.3389/fimmu.2017.00537 PMID: 28539926

17. Ralph DK, Matsen FA IV. Likelihood-based inference of B cell clonal families. PLoS Computational Biology. 2016; 12(10):e1005086. https://doi.org/10.1371/journal.pcbi.1005086 PMID: 27749910

18. Sheng Z, Schramm CA, Connors M, Morris L, Mascola JR, Kwong PD, et al. Effects of Darwinian selection and mutability on rate of broadly neutralizing antibody evolution during HIV-1 infection. PLoS Computational Biology. 2016; 12(5):e1004940. https://doi.org/10.1371/journal.pcbi.1004940 PMID: 27191167

19. Kirik U, Persson H, Levander F, Greiff L, Ohlin M. Antibody Heavy Chain Variable Domains of Different Germline Gene Origins Diversify Through Different Paths. Frontiers in Immunology. 2017; 8. https://doi.org/10.3389/fimmu.2017.01433 PMID: 29180996

20. Cui A, Di Niro R, Vander Heiden JA, Briggs AW, Adams K, Gilbert T, et al. A Model of Somatic Hypermutation Targeting in Mice Based on High-Throughput Ig Sequencing Data. The Journal of Immunology. 2016; 197(9):3566–3574. https://doi.org/10.4049/jimmunol.1502263 PMID: 27707999

21. Honegger A, Pluecthun A. Yet another numbering scheme for immunoglobulin variable domains: an automatic modeling and analysis tool. Journal of Molecular Biology. 2001; 309(3):657–670. https://doi.org/10.1006/jmbi.2001.4662 PMID: 11397087

22. Jaccard P. The distribution of the flora in the alpine zone. New Phytologist. 1912; 11(2):37–50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

23. Tibshirani R. Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological). 1996; 58(1):267–288.

24. Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K. Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2005; 67(1):91–108. https://doi.org/10.1111/j.1467-9868.2005.00490.x

25. Tibshirani RJ. Adaptive piecewise polynomial estimation via trend filtering. The Annals of Statistics. 2014; 42(1):285–323. https://doi.org/10.1214/13-AOS1189

26. Gupta NT, Adams KD, Briggs AW, Timberlake SC, Vigneault F, Kleinstein SH. Hierarchical clustering can identify B cell clones with high confidence in Ig repertoire sequencing data. The Journal of Immunology. 2017; 198(6):2489–2499. https://doi.org/10.4049/jimmunol.1601850 PMID: 28179494

27. Laserson U, Vigneault F, Gadala-Maria D, Yaari G, Uduman M, Vander Heiden JA, et al. High-resolution antibody dynamics of vaccine-induced immune responses. Proceedings of the National Academy of Sciences. 2014; 111(13):4928–4933. https://doi.org/10.1073/pnas.1323862111

28. Vander Heiden JA, Stathopoulos P, Zhou JQ, Chen L, Gilbert TJ, Bolen CR, et al. Dysregulation of B cell repertoire formation in myasthenia gravis patients revealed through deep sequencing. The Journal of Immunology. 2017; 198(4):1460–1473. https://doi.org/10.4049/jimmunol.1601415 PMID: 28087666

29. Stern JN, Yaari G, Vander Heiden JA, Church G, Donahue WF, Hintzen RQ, et al. B cells populating the multiple sclerosis brain mature in the draining cervical lymph nodes. Science Translational Medicine. 2014; 6(248):248ra107–248ra107. https://doi.org/10.1126/scitranslmed.3008879 PMID: 25100741

30. Tsioris K, Gupta NT, Ogunniyi AO, Zimnisky RM, Qian F, Yao Y, et al. Neutralizing antibodies against West Nile virus identified directly from human B cells by single-cell analysis and next generation sequencing. Integrative Biology. 2015; 7(12):1587–1597. https://doi.org/10.1039/c5ib00169b PMID: 26481611

31. Turchaninova MA, Davydov A, Britanova OV, Shugay M, Bikos V, Egorov ES, et al. High-quality full-length immunoglobulin profiling with unique molecular barcoding. Nat Protoc. 2016; 11(9):1599–1616. https://doi.org/10.1038/nprot.2016.093 PMID: 27490633

32. Meng W, Zhang B, Schwartz GW, Rosenfeld AM, Ren D, Thome JJ, et al. An atlas of B-cell clonal distribution in the human body. Nature Biotechnology. 2017; 35(9):879. https://doi.org/10.1038/nbt.3942 PMID: 28829438

33. Shugay M, Britanova OV, Merzlyak EM, Turchaninova MA, Mamedov IZ, Tuganbaev TR, et al. Towards error-free profiling of immune repertoires. Nature Methods. 2014; 11(6):653–655. https://doi.org/10.1038/nmeth.2960 PMID: 24793455

34. Vander Heiden JA, Yaari G, Uduman M, Stern JN, O'Connor KC, Hafler DA, et al. pRESTO: a toolkit for processing high-throughput sequencing raw reads of lymphocyte receptor repertoires. Bioinformatics. 2014; 30(13):1930–1932. https://doi.org/10.1093/bioinformatics/btu138 PMID: 24618469

35. Loman NJ, Misra RV, Dallman TJ, Constantinidou C, Gharbia SE, Wain J, et al. Performance comparison of benchtop high-throughput sequencing platforms. Nature Biotechnology. 2012; 30(5):434–439. https://doi.org/10.1038/nbt.2198 PMID: 22522955

36. Ralph DK, Matsen FA IV. Consistency of VDJ rearrangement and substitution parameters enables accurate B cell receptor sequence annotation. PLoS Computational Biology. 2016; 12(1):e1004409. https://doi.org/10.1371/journal.pcbi.1004409 PMID: 26751373

37. Dunbar J, Deane CM. ANARCI: antigen receptor numbering and receptor classification. Bioinformatics. 2015; 32(2):298–300. https://doi.org/10.1093/bioinformatics/btv552 PMID: 26424857

38. Liao HX, Lynch R, Zhou T, Gao F, Alam SM, Boyd SD, et al. Co-evolution of a broadly neutralizing HIV-1 antibody and founder virus. Nature. 2013; 496(7446):469. https://doi.org/10.1038/nature12053 PMID: 23552890

39. Lefranc MP. Nomenclature of the human immunoglobulin heavy (IGH) genes. Experimental and Clinical Immunogenetics. 2001; 18(2):100–116. https://doi.org/10.1159/000049189 PMID: 11340299

40. Hartigan JA, Wong MA. Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society Series C (Applied Statistics). 1979; 28(1):100–108.

41. Hastie T, Tibshirani R, Tibshirani RJ. Extended Comparisons of best subset selection, forward stepwise selection, and the lasso. arXiv preprint arXiv:170708692. 2017;.

42. Boyd SP, Vandenberghe L. Convex Optimization. Cambridge University Press; 2004.

43. Byrd RH, Lu P, Nocedal J, Zhu C. A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing. 1995; 16(5):1190–1208. https://doi.org/10.1137/0916069

44. Tomlinson IM, Cox J, Gherardi E, Lesk A, Chothia C. The structural repertoire of the human V kappa domain. The EMBO journal. 1995; 14(18):4628–4638. https://doi.org/10.1002/j.1460-2075.1995.tb00142.x PMID: 7556106

45. Breden F, Luning EP, Peters B, Rubelt F, Schramm C, Busse C, et al. Reproducibility and Reuse of Adaptive Immune Receptor Repertoire Data. Frontiers in Immunology. 2017; 8:1418–1418. https://doi.org/10.3389/fimmu.2017.01418 PMID: 29163494

46. Rubelt F, Busse CE, Bukhari SAC, Bürckert JP, Mariotti-Ferrandiz E, Cowell LG, et al. Adaptive Immune Receptor Repertoire Community recommendations for sharing immune-repertoire sequencing data. Nature Immunology. 2017; 18(12):1274. https://doi.org/10.1038/ni.3873 PMID: 29144493

47. Jones DT, Buchan DW, Cozzetto D, Pontil M. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics. 2011; 28(2):184–190. https://doi.org/10.1093/bioinformatics/btr638 PMID: 22101153

48. Ekeberg M, Lövkvist C, Lan Y, Weigt M, Aurell E. Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. Physical Review E. 2013; 87(1):012707. https://doi.org/10.1103/PhysRevE.87.012707

49. Wang N, Smith WF, Miller BR, Aivazian D, Lugovskoy AA, Reff ME, et al. Conserved amino acid networks involved in antibody variable domain interactions. Proteins: Structure, Function, and Bioinformatics. 2009; 76(1):99–114. https://doi.org/10.1002/prot.22319