Low Rank Smoothed Sampling Methods for Identifying Impactful Pair-wise Mutations

Nicholas Majeske
Computer Science
Western Washington Univ.
Bellingham, WA
Majeskn@wwu.edu

Filip Jagodzinski
Computer Science
Western Washington Univ.
Bellingham, WA
Filip.Jagodzinski@wwu.edu

Brian Hutchinson*
Computer Science
Western Washington Univ.
Bellingham, WA
Brian.Hutchinson@wwu.edu

Tanzima Islam
Computer Science
Western Washington Univ.
Bellingham, WA
Tanzima.Islam@wwu.edu

***Abstract*—Proteins are involved in nearly all biophysical processes. Even a single amino acid substitution in a protein can render a biomolecule inoperative, else be the cause of a debilitating disease. Experimentally studying the effects of all possible mutations in a protein is infeasible since it requires a combinatorial number of mutants to be engineered and analyzed. Computational methods developed for studying the impact of single amino acid substitutions do not scale in handling the number of mutants that are possible for two amino acid substitutions. In this work, we present an approach for reducing the amount of mutation samples that need to be used to predict the impact of pair-wise amino acid substitutions. We evaluate the effectiveness of the proposed approach by – (1) generating exhaustive mutations in silico for 8 proteins with 2 amino acid substitutions, (2) analyzing the mutants via rigidity analysis, and producing *ground truth* mutation data, and (3) comparing the predictions from the sampled method to that in the ground truth dataset. We show that it is possible to approximately predict the effect of the two amino acid substitutions using as low as 25%d of all mutations, and that these approximations are improved by imposing a low rank constraint.**

***Keywords*—protein, mutations, big data, sampling, reduction**

I. INTRODUCTION

Inferring the effects of amino acid substitutions has a wide range of applications in the biochemical sciences. Knowing the extent to which a mutation alters a protein's stability can aid in drug design studies that aim to deliver pharmaceutical solutions for combating diseases caused by protein mutants [1].

One approach to infer the effect of a mutation in the physical protein is to conduct a free energy of unfolding experiment by denaturing a protein mutant and its non-mutated form (wild type). The extent to which the wild type denatures relative to the mutant is used by the Schellman equation to provide a $\Delta\Delta G$ measurement (change of Gibbs free-energy) offering a quantitative assessment of the effect of the mutation(s) [2].

Unfortunately, mutation studies performed in the physical protein are time and cost prohibitive. Performing even a small subset of all possible mutations in a wet lab setting and experimentally inferring the effects of those amino acid substitutions might require months of work.

To help complement and inform wet lab work, modeling and computational methods are available. They strive to predict the effects of mutations, with varying degrees of accuracy. Early approaches searched for best side-chain conformations as a measure of the impact of a mutation [3]–[5], and relied on heuristic energy functions or database-derived potentials [6], [7]. Other approaches are dependent on sufficiently large datasets of homologous proteins [8]–[10]. Machine learning (ML) approaches, which is a branch of artificial intelligence, have also been leveraged to infer the effects of mutations. Some rely on Vector Machine methods [11], [12], while others utilize Random Forest and similar approaches [13], [14]. Among these ML methods, several have high prediction accuracy rates (upwards of 80%) of the effects of mutations involving single amino acid substitutions.

MOTIVATION AND CONTRIBUTIONS

Energy-, homology-, and ML-based approaches for inferring the effects of mutations have several drawbacks. All but a few of them permit reasoning about the effects of single point mutations only [9], [15], [16], [16]–[22]. But there is a clear need to understand the effects of multiple mutations. For example, for HIV-1 protease it has been shown that the median number of mutations in the protease gene which confers drug-associated resistance to protease inhibitors duranavir and tipranavir is twenty-eight [23].

Unfortunately free energy changes for single mutations cannot be summed to predict the effect of performing those mutations all at once. There are several such instances in the literature and ProTherm [24], a database of mutation experiments done in the wet lab. For example, the single W94L mutation in *Barnase Bacillus amyloliquefaciens* yields a $\Delta\Delta G$ of -1.59 (ProTherm entry 2262), and the single H18G mutation yields a $\Delta\Delta G$ of -0.98 (ProTherm entry 2263). These two sum to -1.59 + -0.98 = -2.57. However, when both mutations are performed at the same time in the physical protein, the experimental $\Delta\Delta G$ value is -1.17 (ProtHerm entry 2264).

Many computational approaches also do not provide details about the extent of a mutation. For example work by Gohlke [25], which relies on rigidity analysis, outputs a single all-atom cluster configuration entropy value, but does not permit reasoning at the residue-level about the effect of the amino acid substitutions. Eris [26] summarizes that existing computational methods predict the general trend of free energy change upon mutation, but that they fail in providing details.

*Brian Hutchinson has a joint appointment with the Computing and Analytics Division of Pacific Northwest National Laboratory, Richland, WA.

For this work, we are motivated by a need to explore which pairs of mutations have an impact on a protein's structure. Due to the vast number of possible mutants with two amino acids substitutions that can be engineered for even a small protein – for a 99 residue biomolecule, for example, 1,751,211 unique mutants are possible – this is a big data problem that even for efficient computational approaches becomes intractable. Our contributions are two-fold.

Firstly, we have engineered a software suite for generating mutants with 2 amino acid substitutions, and generate an exhaustive set of possible mutants for each of 8 proteins. We perform a quick analysis of the rigid and flexible regions of the *in silico* generated mutant and wild type structures using an efficient graph theoretic algorithm, and rely on our past rigidity metric scores to infer the effects of the mutations. These exhaustive results are treated as the **ground truth** about the effects of the amino acid substitutions.

Secondly, because performing such exhaustive studies is computationally intensive, we present methods to accurately approximate the ground truth using a fraction of the total samples. In general, the fewer samples these **empirical models** are based upon, the more computationally efficient they will be, but at the expense of approximation quality. To counteract the effect of random noise on the empirical models, we employ a smoothing technique based on matrix rank, yielding **low rank** estimates that are able to filter out noise and improve approximation quality.

II. Related Work

Here we survey existing computational approaches for inferring the effects of mutations, and overview low rank factorization.

The majority of computational approaches for inferring the effects of mutations reason about the impact of single amino acid substitutions. PoPMuSiC 2.1 [27] makes predictions about $\Delta\Delta$G and generates a sequence optimality score. AutoMute [28] is a ML-based method that requires a large training set. CUPSAT [29] relies on energy potentials (atomic and torsional angles), requires calculating Boltzmann's energy values, and is dependent on a radial pair distribution function, whose calculation is time intensive. D-Mutant [30] constructs a residue-specific all-atom potential and requires the use of 1,011 actual protein structures with 2Å resolution or better. I-mutant2.0 is an SVM-based tool that correctly predicts (with a cross-validation procedure) 80% or 77% of the data set, depending on the usage of structural or sequence information, respectively. The input is a single PDB [31] file, along with a chain ID and residue number [32]. STRUM [33] is a physics-based energy calculation approach that relies on multiple-threading template alignment. McCafferty [34] has developed an unfolding mutation screen (UMS) that relies on residue propensity tables and calculates free energy changes.

Of the few approaches that permit reasoning about the effects of multiple mutations, none are able to perform sreening-like analyses. MAESTRO and MAESTROweb [35] are a machine learning based approach for predicting $\Delta\Delta G$ values for single and multiple mutations, but do not permit a screening of all possible multiple-mutation variants. mCSM [36], too, permits inferring the effects of mutations, and predicts a $\Delta\Delta G$ value, but it does not allow a user to perform a screen in which a subset of pairwise mutations are assessed. DUET [37] which consolidates mCSM and SDM, another prediction tool, relies on a support vector machine approach to study missense mutations. It, too, makes a prediction about $\Delta\Delta G$ but does not allow identifying which pairwise mutations are impactful.

In our most recent work, we have developed a compute pipeline for generating in silico all mutants with pairwise mutations [38]. Our Allostery Impact Map infographic aids to identify residues that when mutated along with another amino acid cause a disruption to the protein's stability as inferred using rigidity analysis.

Low rank matrix factorization is at the heart of a wide range of data analysis techniques [39]; for example, the popular principal component analysis dimensionality reduction technique [40]. Low rank is often found in matrices describing interactions between two entities. One famous example is the *Netflix* movie recommendation problem [41], [42], where the goal is to predict what rating a given user would assign to a given movie. The low rank property arises because the inherent dimension of the interaction is significantly smaller than the ambient dimension. For example, there are *n kinds of users* and *m kinds of movies*, and the interactions between them explain much of the ratings in user-movie ratings.

Other real world problems have exploited the low rank structure, from image compression [43] to syntactic models of natural language [44], [45]. As discussed in Section III-C, in this work we explore low rank structure in matrices describing the effects of double mutations in proteins. We use a singular value decomposition [46] to find a low rank approximation based on the well-known Eckart-Young-Mirksy theorem [47].

III. Methods

Exhaustive mutation sets have been used in the past to explore and identify impactful amino acid substitutions [48]. However, generating all possible mutants with 2 amino acid substitutions can take several weeks – even months – of compute time. For example, the exhaustive pair-wise mutation dataset generated in [38] using 8 compute cores took upwards of a week for pdb file 2lzm, of a 164 residue lysozyme.

In this paper, we present a multi-phase compute pipeline (Figure 1) for evaluating several sampling methods to reduce the amount of data used to make good predictions about the effects of pairwise mutations. Such reduction in data improves the scalability of the prediction operation. We also evaluate

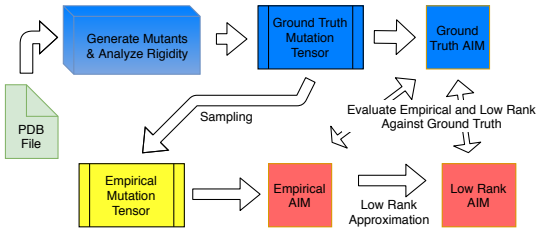


Fig. 1: Pipeline for producing ground truth (Phase 1, blue), sampling (Phase 2, yellow), and generating empirical and low rank Allostery Impact Maps (AIMs) (Phase 3, red).

the quality of the proposed sampling methods by quantifying how close the predictions are with respect to the exhaustive mutation set. The pipeline is comprised of three phases:

- ***Phase 1: Generating ground truth*** – This phase is required for validating the effectiveness of our sampling methods. There are two tasks: (i) generating the exhaustive set of all possible mutants having two amino acid substitutions, and (ii) analyzing the effects of the mutations using rigidity analysis.
- ***Phase 2: Sampling from ground truth*** – We apply our proposed sampling methods to specifically study the impact of pair-wise mutations to hydrophobic, hydrophilic, and mutations sampled at random.
- ***Phase 3: Low Rank smoothing*** – To improve the approximation quality of the sampled (empirical) Allostery Impact Maps (AIMs), we impose a low rank constraint, producing a low rank AIM. Smoothing the empirical AIMs reduces noise and thus improves approximation quality.

We explain each of these phases, as well as details of the tasks involved, in the following subsections.

A. Phase 1: Generating the Ground Truth Data

Generating Mutant Structures: We rely on our ProMuteHT software for generating mutants *in silico* [49]. It relies on a variety of homology modeling, as well as energy minimization MD runs, via integration with custom scripts tools such as NAMD and SCWRL 4.0 [50]–[52].

Rigidity Analysis: Rigidity analysis [53]–[55] is a graph-based method that provides information about the rigid regions of biomolecules [56]. Atoms and their chemical interactions are used to construct a mechanical model of a molecule, for which a graph is constructed and analyzed using pebble game algorithms [57]. The results are used to infer the rigid regions of the protein, identified as rigid clusters of atoms (Figure 2). We use rigidity analysis because analysis of a several hundred amino acid protein takes only a handful of seconds.

For this work, we tally the counts and distribution of rigid clusters in the wild type, as well as a mutant, to quantitatively assess the effect of the amino acid substitutions performed *in silico*. We use the following rigidity metric (see [48]) :

$$RD_{WT \rightarrow mutant} : \sum_{i=1}^{i=LRC} i \times [WT_i - Mut_i] \qquad (1)$$

where WT refers to Wild Type, Mut refers to mutant, and LRC is the size of the Largest Rigid Cluster (in atoms). Each summation term of $RD_{WT \rightarrow mutant}$ calculates the difference in the count of a specific cluster size, i, of the wild type and mutant, and weighs that difference by i.

Allostery Impact Map: We use the rigidity analysis data to create a **Ground Truth Mutation Tensor**, $\mathcal{T}^{gt} \in \mathbb{R}^{n \times n \times 361}$. The $(i, j, k)^{th}$ element, $\mathcal{T}^{gt}_{ijk}$, contains the rigidity data for performing the kth pair of substitutions (out of $19^2 = 361$ total possible pairs of substitutions) at residues i and j.

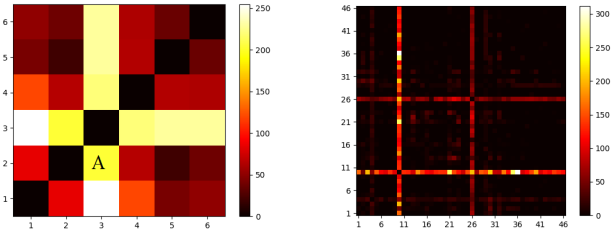
From $\mathcal{T}^{gt}$, we build a Ground Truth Allostery Impact Map (AIM), M^{gt}, [38] which provides a visual infographic based on quantitative data for reasoning about the effects of mutating two residues. Figure 3 shows two sample AIMs, in which the $x-$ and $y-$axis values designate amino acids in the chain of residues in a protein. The color of any one cell in the AIM designates a sum value of all the metrics for all of the 361 mutant structures when the residues indicated by the x and y values are exhaustively mutated.

(a) cartoon (b) rigidity results

Fig. 2: The output of rigidity analysis for PDB file 1csp (a) indentifies atoms belonging to the same rigid clusters (b).

Because of the large count of structures that make up an exhaustive set of all pairwise mutations for a protein, we distribute the computational tasks for Phase 1 among 165 compute cores. Each core further subdivided each computational task via process-level parallelism by spawning 1 mutex process for mutation for each available core in a given machine. With these resources, we achieved a process-level granularity of $19^k \binom{n}{k} / (165 * 8)$ when generating all possible protein mutants containing k mutation sites from a wildtype n amino acids.

In this phase, our compute pipeline leverages the knowledge that no two pair-wise protein mutations depend on each other to parallelize the chain of tasks – mutation generation, rigidity analysis, and AIM generation for each mutation. These mutually independent computation tasks mutating pair-wise substitutions can be run in a distributed computing environment using local storage space on each compute machine. The use of parallel computing ensures that the Phase 1 finishes fast and that of local storage space ensures that these I/O bound tasks (since involves file I/O) do not overwhelm the network or the attached network filesystem.



(a) Example (b) AIM for pdb 1crn

Fig. 3: Allostery Impact Map : The color of a cell specifies the sum values for our rigidity metric for all 361 mutants generated by exhaustively mutating the amino acids indicated by the x and y axis values. The cell marked A (a), at $x = 3$, $y = 2$, is the sum metric for all 361 mutants for when residues $i = 3$ and $j = 2$ were exhaustively mutated. (b) is reproduced from [38]

B. Phase 2: Generating the Empirical Allostery Impact Map

In estimating the Allostery Impact Map, we use three methods of sampling to derive three different empirical AIMs. These three methods include: sampling randomly from the set of all mutations (True Random, denoted $\mathcal{T}_{tr}^{emp}$), sampling randomly from the set of mutations in which all substitutions are to hydrophobic amino acids (Mutation to Hydrophobic, denoted $\mathcal{T}_{pho}^{emp}$), and sampling randomly from the set of mutations in which all substitutions are to hydrophilic amino acids (Mutation to Hydrophilic, denoted $\mathcal{T}_{phi}^{emp}$).

For each of these three sampling methods, we analyze the quality of approximation as a function of the quantity of sampling in the following two ways. First, we sweep the number of mutation site pairs being sampled in {25%, 50%, 75%, 100%} while holding the number of mutations sampled for each of these site pairs constant at 19. Additionally, for empirical AIMs where less than 100% of mutation site pairs are sampled, we have an 'unfilled' empirical where mutation site pairs are left unsampled, and a 'filled' empirical where all unsampled mutation site pairs are set to the average metric of all sampled mutation site pairs. Second, we sweep the number of mutations sampled for each mutation site pair in {5%, 10%, ..., 95%, 100%} while holding the number of mutation site pairs constant at 100%.

An important distinction to make in these sampling methods is that the sample space for $\mathcal{T}_{pho}^{emp}$ and $\mathcal{T}_{phi}^{emp}$ at a given mutation site pair is significantly different from the sample space of $\mathcal{T}_{tr}^{emp}$. In this work, we consider the following to be the set of hydrophobic (pho) and hydrophilic ($phil$) amino acids (abbreviations):

pho	Ala, Gly, Val, Leu, Iso, Pro, Phe, Met, Trp
$phil$	Tyr, Asn, Cys, Gln, Ser, Thr, Asp, Glu, Arg, His, Lys

To facilitate exploring if sampling from mutations to hydrophobic residues or sampling from mutations to hydrophilic residues has any bearing on the quality of the results when compared to sampling among any type of mutation, we define MT_{pho} and MT_{phil}. Let $WTSeq$ be the amino acid sequence of length n for a wildtype protein and $WTSeq_i$ denote the i_{th} amino acid of that sequence for $i \in \{1, 2, ..., n\}$. Additionally, let M denote the set of mutation sites at which amino acid substitutions have been made for a given protein mutation. Note that $\|M\| = k$ and $k = 2$ in this work.

$$MT_{pho} = \prod_{i \in M} f_{pho}(WTSeq_i) \text{ where } f_{pho}(x) = \begin{cases} 8 & x \in pho \\ 9 & x \in phil \end{cases}$$

$$MT_{phil} = \prod_{i \in M} f_{phil}(WTSeq_i) \text{ where } f_{phil}(x) = \begin{cases} 10 & x \in phil \\ 11 & x \in pho \end{cases}$$

Thus, for $k = 2$ the number of possible mutations to hydrophobic and hydrophilic for given mutation site pair M are $MT_{pho} \in [64, 81]$ and $MT_{phil} \in [100, 121]$ respectively. For for $k = 2$, the number of possible mutations for any mutation site pair M is $19^2 = 361$, meaning that MT_{pho} and MT_{phil} account for up to 22.4% and 33.5% of the set of mutations site pair M.

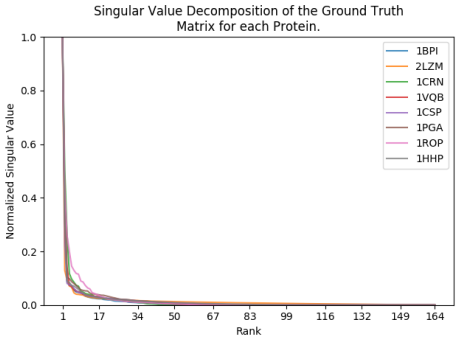


Fig. 4: Singular values for the eight proteins, revealing approximately low rank structure. To aid comparison, singular values have been normalized by dividing by the largest singular value.

In this phase, our pipeline again leverages a distributed computing environment to apply our sampling methods. We are able to do so because the sampling methods can run independently of the others. To ensure scalability of our compute pipeline, a separate process on a computing node applies a sampling method to generate a subset of mutations. Hence, the overhead of evaluating several sampling methods is only bound by the slowest running method. Within each process (that handles a specific sampling method), we leverage the knowledge that mutations in different site locations can be generated in parallel, and hence divide these tasks across available cores on a computing node. Our use of scripting languages BASH and Python ensures that the same job submission and management scripts can be used to run our pipeline on a distributed computing environment such as Condor.

C. Phase 3 : Generating Low Rank Allostery Impact Map

While our empirical AIM is fast to generate, it by definition paints an incomplete picture of the ground truth AIM. "Filling in" the missing information requires making some assumption about global structure of the ground truth matrix. In our case, we assume that the ground truth matrix is *low rank*. The rank of a matrix is the number of linearly independent columns (and rows) in the matrix; equivalently, it is defined as the number of non-zero singular values. Rank can be thought of as a notion of complexity in the matrix: low rank matrices can be explained by a relatively small number of underlying factors. Figure 4 plots the singular values (in the conventional descending order) for the ground truth AIMs for all of the proteins considered. While none of the matrices are exactly low rank, all are approximately low rank: most of the singular values are approximately zero.

If we let M^{emp} be the empirical AIM, our low rank matrix is the solution to the following convex optimization problem:

$$\arg\min_{M} \quad \|M^{emp} - M\|_F \tag{2}$$

$$s.t. \quad \text{rank}(M) \leq R \tag{3}$$

where R is the desired rank (a value to be assessed empirically). The famous theorem of Eckert-Young-Mirsky states that

the closed form solution to this problem is:

$$M_R^{emp} = U\Sigma_R V^T. \quad (4)$$

Here U and R are the left and right singular values of M^{emp}, respectively, and Σ is the matrix whose diagonal contains the singular values of M^{emp}; all three matrices can be obtained by a singular value decomposition. Σ_R is Σ with all but the R largest singular values replaced by zeros. Our low rank AIM, M^{lr} is defined to be M_R^{emp}, the optimal rank R approximation of M^{emp}. Note that this assumes we want to approximate M^{emp} at all sites, which is suboptimal when using a sampling strategy that does not sample all sites. Despite this limitation, our experiments show that the approach works well, and we leave weighted approximations [58] for future work.

D. Data Preparation

We generated all possible mutants with two amino acid substitutions for 8 proteins (Table I). They varied in size from the 46 residue PDB file 1crn of the protein crambin, to the 164 residue PDB file 2lzm of bacteriophage T4 lysozyme.

TABLE I: PDB files used, and mutants generated

PDB file	num residues	mutants	runtime
1crn	46	373,635	23m
1pga	56	555,940	37m
1bpi	58	596,733	42m
1rop	63	705,033	51m
1csp	67	798,171	1.1h
1vqb	87	1,350,501	1.5h
1hhp	99	1,751,211	2.6h
2lzm	164	4,825,126	8.9h

E. Evaluation Metrics

We evaluate the quality of approximation using the Sum of Absolute Error (SAE) for the ground truth AIM:

$$SAE = \sum_{i=1}^{n}\sum_{j=1}^{n} |M_{ij}^{gt} - M_{ij}| \quad (5)$$

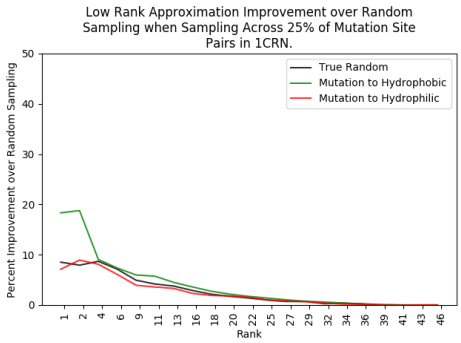
where M is either an empirical AIM, M^{emp}, or a low rank AIM, M^{lr}. As the number of samples increases, M^{emp} approaches M^{gt} and the SAE approaches zero.

IV. Results - Case Studies

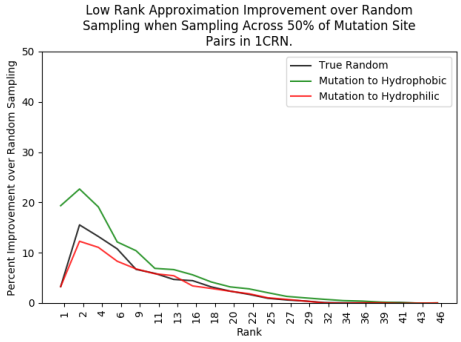
In this section, we evaluate the efficacy of our low rank smoothed sampling methods by – (a) computing the SAE compared to ground truth over empirical approximation (the lower the better), and (b) measuring how accurately a significantly reduced subset of the exhaustive mutation set can reconstruct the characteristic bands (indicating mutation sensitive sites).

A. Low Rank Improvement over Random Sampling

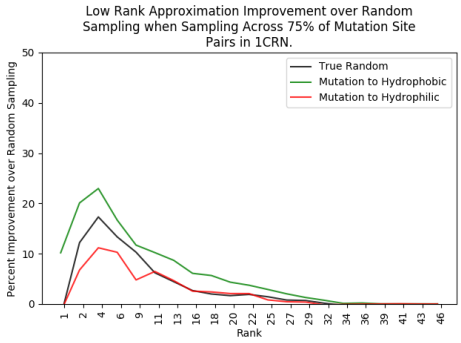
Figure 5 shows that the low rank model consistently reduces SAE relative to the empirical model on 1crn for small values of the rank, R. As R approaches 46, M^{lr} approaches M^{emp} and the improvement converges to 0. The biggest improvements by smoothing are in the "Mutation to Hydrophobic" case, suggesting this subset of the data is particularly well-suited to the low rank assumption. In contrast, the "Mutation to Hydrophilic" benefits the least from the smoothing.



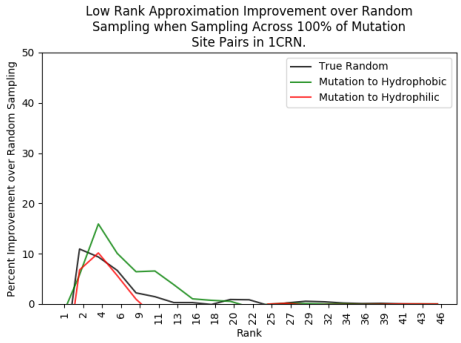
(a)



(b)



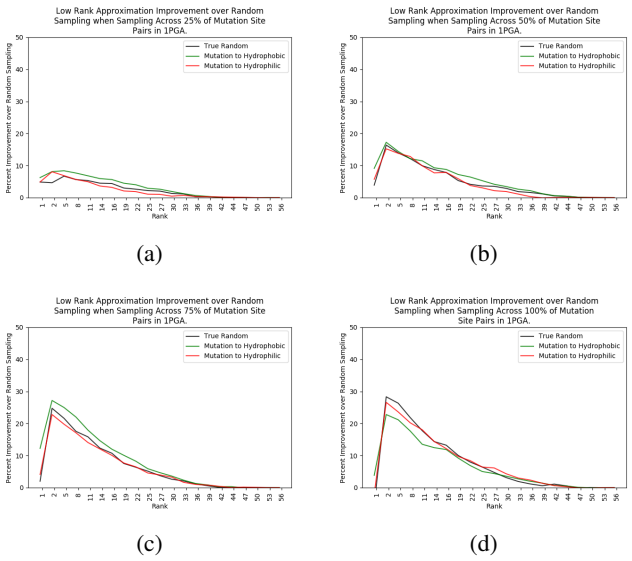
(c)



(d)

Fig. 5: Improvement in SAE by low rank smoothing relative to the "filled" empirical approximation when randomly sampling 19 mutations across mutation site pair sampling at increments of 25% for 1crn.

Figure 6 shows analysis of 1pga. Unlike 1crn, we see a distinctive increase in the improvement as the fraction of sites sampled approaches 1.0, achieving a relative reduction in SAE of up to 34% (i.e. matches closely). This indicates that all mutation sites encode unique information as opposed to 1crn where most information is encoded in a small number of sites. This shows that our study enables us to compare the characteristics of mutation sites across different proteins.



(a) (b)

(c) (d)

Fig. 6: Improvement in SAE by low rank smoothing relative to the "filled" empirical approximation when randomly sampling 19 mutations across mutation site pair sampling at increments of 25% for 1pga.

B. Low Rank Approximation and Random Sampling Error

Figure 7 plots the absolute error (SAE) for the empirical and low rank models for the three sampling types (Y axis) across ranks (along X axis), using the "unfilled" sampling strategy on 2lzm. For this protein, there is a clear basin of good values of R ranging from 8-32. As expected, SAE of our low rank approximation approaches to the empirical as it approaches full rank.

Figure 8 repeats the previous analysis, with sampling from 75% of pair-wise mutation sites for 1hhp. Again a clear "sweet spot" for the low rank approximation is evident, this time at a much lower rank. Interestingly, the error is much higher for "to hydrophobic" than "to hydrophilic" in this case. This might be explained biophysically, because mutating a residue from a hydrophilic to a hydrophobic one would render an otherwise content surface exposed residue to become energetically unfavorable. We leave an assessment exploring the hydrophobicity and surface accessible attributes as affecting sampling rates to future studies.

C. Heatmaps

Figures 9 and 10 present M^{gt}, M^{emp} and M^{lr} for 2lzm and 1crn, respectively. In both cases, the low rank model is able to detect the banded, low rank structure of the ground

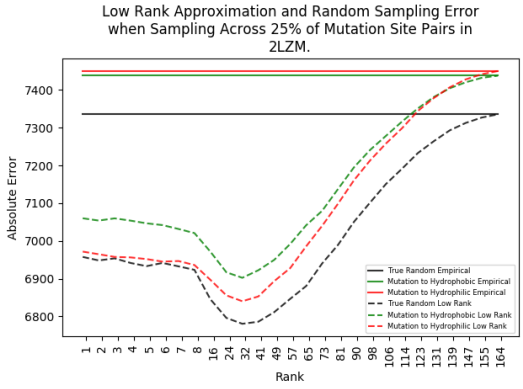


Fig. 7: 2lzm : Empirical approximation error against low rank approximation error for various ranks when sampling across 25% of mutation site pairs.

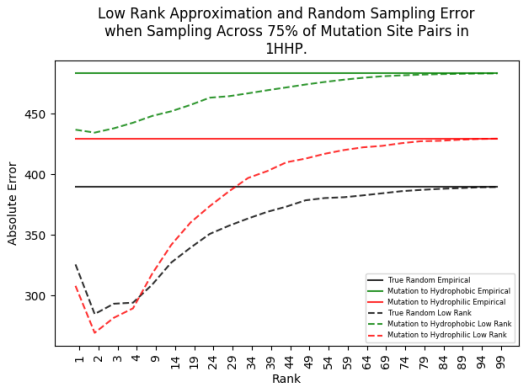


Fig. 8: 1hhp : Error rates of random sampling empirical heatmap against low rank approximation for various ranks when sampling across 75% of mutation site pairs.

truth matrix from the samples in the empirical model. In the case of 1crn, it appears to over-generalize.

V. CONCLUSIONS & FUTURE WORK

We have developed a software suite for generating mutants with 2 amino acid substitutions with the aim of motivating a computational approach for identifying impactful pairs of mutations. We have exhaustively generated mutant sets for 8 proteins, and analyzed both the wild type and mutants using rigidity analysis; we call this exhaustive analysis the ground truth. Because even computational approaches for such exhaustive screens are time consuming, we have presented several methods to accurately approximate the ground truth using a fraction of the total samples from the exhaustive data.

We observed several interesting results when comparing the ground truth, empirical, and low rank approximations among our case studies. In some proteins – 2lzm for example – prediction accuracy was sensitive to random noise in the data. In those cases, a large rank was needed to smooth out the noise when sampling only 25% of ground truth mutations. We found that for some proteins – 1pga for example – the

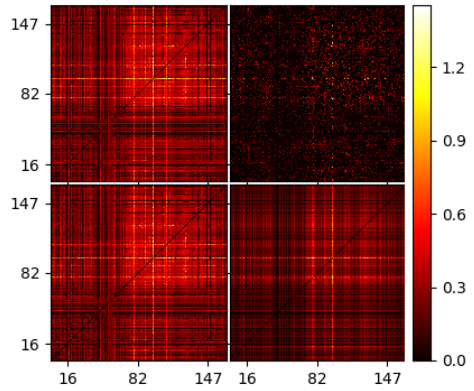


Fig. 9: 2lzm : Ground truth (left), empirical approximation (upper right), and low rank approximation (lower right).

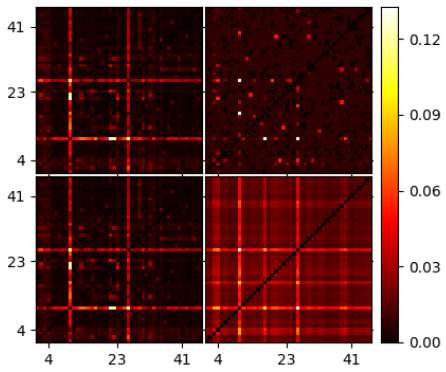


Fig. 10: 1crn : Ground truth (left), empirical approximation (upper right) and rank approximation (low right).

mutation sites encode unique information, but for others such as 1crn, most information about the effects of mutations was encoded in a small number of sites. The fact that a choice of a sampling rate, and choice of the specific type of sampling (whether from mutations to hydrophobic, or sampling from mutations to hydrophilic residues) results in different low rank approximations for different proteins suggests that any one sampling strategy is not generalizable for all biomolecules. We leave to future studies an exploration of the interplay of sampling strategies in combination with structural and classification properties of molecules in attaining empirical and low rank based predictions with good approximations to the ground truth for all proteins.

There are several ways the low rank modeling part of this work could be extended. First, weighted low-rank decompositions [58] would likely improve the quality of the low rank approximation. While our approximation optimization problem is convex, rank minimization problems are in general non-convex; for more sophisticated formulations we would likely need to consider convex relaxations of rank [59]. Finally, it would be worth exploring low rank decompositions explicitly designed to be robust to noise [60].

REFERENCES

[1] B. Reva, Y. Antipin, and C. Sander, "Predicting the functional impact of protein mutations: application to cancer genomics," *Nucleic Acids Research*, 2011.

[2] J. A. Schellman, "The thermodynamic stability of proteins," *Annual review of biophysics and biophysical chemistry*, vol. 16, no. 1, pp. 115–137, 1987.

[3] R. J. Dunbrack and M. Karplus, "Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains," *Nature Structural Biology*, vol. 1, pp. 334–340, 1994.

[4] J. Janin and S. Wodak, "Conformation of amino acid side-chains in proteins." *J Mol Biol*, vol. 125, no. 3, pp. 357–386, Nov 1978.

[5] J. Ponder and F. Richards, "Tertiary templates for proteins: Use of packing criteria in the enumeration of allowed sequences for different structural classes," *Journal Molecular Biology*, vol. 193, pp. 775–791, 1987.

[6] D. Gilis and M. Rooman, "Predicting protein stability changes upon mutation using database-dervied potentials: Solvent accessibility determines the importance of local versus non-local interactions along the sequence," *Journal of Molecular Biology*, vol. 272, no. 2, pp. 276–290, 1997.

[7] C. Lee and M. Levitt, "Accurate prediction of the stability and activity effects of site-directed mutagenesis on a protein core," *Nature*, vol. 352, pp. 448–451, 1991.

[8] C. Topham, N. Srinivasan, and T. Blundell, "Prediction of the stability of protein mutants based on structural environment-dependent amino acid substitutions and propensity tables," *Protein Engineering*, vol. 10, no. 1, pp. 7–21, 1997.

[9] C. Worth, R. Preissner, and L. Blundell, "Sdm-a server for predicting effects of mutations on protein stability and malfunction," *Nucleic Acids Research*, vol. 39, no. Web Server Issue, pp. W215–W222, 2011.

[10] J. R. Brender and Y. Zhang, "Predicting the effect of mutations on protein-protein binding interactions through structure-based interface profiles," *PLoS Comput Biol*, vol. 11, no. 10, p. e1004494, 2015.

[11] J. Cheng, A. Randall, and P. Baldi, "Prediction of protein stability changes for single-site mutations using support vector machines," *PROTEINS: Structure, Function, and Bioinformatics*, vol. 62, pp. 1125–1132, 2006.

[12] F. Jagodzinski, B. Akbal-Delibas, and N. Haspel, "An evolutionary conservation & rigidity analysis machine learning approach for detecting critical protein residues," in *CSBW (Computational Structural Bioinformatics Workshop), in proc. of ACM-BCB (ACM International conference on Bioinformatics and Computational Biology)*, September 2013, pp. 780–786.

[13] L. Jia, R. Yarlagadda, and C. C. Reed, "Structure based thermostability prediction models for protein single point mutations with machine learning tools," *PloS one*, vol. 10, no. 9, p. e0138022, 2015.

[14] Y. Li and J. Fang, "Prots-rf: a robust model for predicting mutation-induced protein stability changes," *PloS one*, vol. 7, no. 10, p. e47247, 2012.

[15] E. Capriotti, P. Fariselli, and R. Casadio, "A neural-network-based method for predicting protein stability changes upon single point mutations," *Bioinformatics*, vol. 20, Supplemental, pp. i63–i68, 2004.

[16] W. Lee, P. Yue, and Z. Zhang, "Analytical methods for inferring functional effects of single base pair substitutions in human cancers," *Human Genetics*, vol. 126, no. 481-498, 2009.

[17] S. Mooney, "Bioinformatics approaches and resources for single nucleotide polymorphism functional analysis," *Briefings in Bioinformatics*, vol. 6, pp. 44–56, 2005.

[18] S. Henikoff and P. Ng, "Predicting the effects of amnio acid substitutions on protein functions," *Annual Reviews of Genomics Human Genetics*, vol. 7, pp. 61–80, 2006.

[19] S. Teng, E. Michonova-Alexova, and E. Alexov, "Approaches and resources for prediction of the effects of non-synonymous single nucleotide polymorphisms on protein function and interactions," *Current Pharmacology Biotechnology*, vol. 9, pp. 123–133, 2008.

[20] C. Topham, N. Srinivasan, and T. Blundell, "Prediction of the stability of protein mutants based on structural environment-dependent amino acid substitution and propensity tables," *Protein Engineering*, vol. 10, pp. 7–21, 2012.

[21] M. Masso and I. I. Vaisman, "Structure-based prediction of protein activity changes: assessing the impact of single residue replacements," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 3221–3224.

[22] E. H. Kellogg, A. Leaver-Fay, and D. Baker, "Role of conformational sampling in computing mutation-induced changes in protein structure and stability," *Proteins: Structure, Function, and Bioinformatics*, vol. 79, no. 3, pp. 830–838, 2011.

[23] S.-Y. Rhee, J. Taylor, W. J. Fessel, D. Kaufman, W. Towner, P. Troia, P. Ruane, J. Hellinger, V. Shirvani, A. Zolopa, and R. W. Shafer, "Hiv-1 protease mutations and protease inhibitor cross-resistance," *Antimicrobial Agents and Chemotherapy*, vol. 59, no. 8, pp. 4253–4261, 2010.

[24] K. A. Bava, M. M. Gromiha, H. Uedaira, K. Kitajima, and A. Sarai, "Protherm, version 4.0: thermodynamic database for proteins and mutants," *Nucleic acids research*, vol. 32, no. suppl 1, pp. D120–D121, 2004.

[25] S. Radestock and H. Gohlke, "Exploiting the link between protein rigidity and thermostability for data-driven protein engineering," *Engineering in Life Sciences*, vol. 8, no. 5, pp. 507–522, 2008.

[26] V. Potapov, M. Cohen, and G. Schreiber, "Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details," *Protein Engineering Design and Selection*, vol. 22, no. 9, pp. 553–560, 2009.

[27] Y. Dehouck, J. Kwasigroch, M. Gilis, and R. M, "Popmusic 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality," *BMC Bioinformatics*, vol. 12, 2011.

[28] M. Masso and I. I. Vaisman, "Auto-mute: web-based tools for predicting stability changes in proteins due to single amino acid replacements," *Protein Engineering Design and Selection*, vol. 23, no. 8, pp. 683–687, 2010.

[29] V. Parthiban, M. M. Gromiha, and D. Schomburg, "Cupsat: prediction of protein stability upon point mutations," *Nucleic Acids Research*, vol. 34, no. suppl 2, pp. W239–W242, 2006.

[30] H. Zhou and Y. Zhou, "Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction," *Protein science*, vol. 11, no. 11, pp. 2714–2726, 2002.

[31] F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, "The protein data bank," *European Journal of Biochemistry*, vol. 80, no. 2, pp. 319–324, 1977.

[32] E. Capriotti, P. Fariselli, and R. Casadio, "I-mutant2. 0: predicting stability changes upon mutation from the protein sequence or structure," *Nucleic acids research*, vol. 33, no. suppl 2, pp. W306–W310, 2005.

[33] L. Quan, Q. Lv, and Y. Zhang, "Strum: structure-based prediction of protein stability changes upon single-point mutation," *Bioinformatics*, vol. 32, no. 19, pp. 2936–2946, 2016.

[34] C. L. McCafferty and Y. V. Sergeev, "In silico mapping of protein unfolding mutations for inherited disease," *Scientific Reports*, vol. 6, p. 37298, 2016.

[35] J. Laimer, H. Hofer, M. Fritz, S. Wegenkittl, and P. Lackner, "Maestro-multi agent stability prediction upon point mutations," *BMC bioinformatics*, vol. 16, no. 1, p. 116, 2015.

[36] D. E. Pires, D. B. Ascher, and T. L. Blundell, "mcsm: predicting the effects of mutations in proteins using graph-based signatures," *Bioinformatics*, vol. 30, no. 3, pp. 335–342, 2013.

[37] ——, "Duet: a server for predicting effects of mutations on protein stability using an integrated computational approach," *Nucleic acids research*, vol. 42, no. W1, pp. W314–W319, 2014.

[38] N. Majeske and F. Jagodzinski, "Elucidating which pairwise mutations affect protein stability: An exhaustive big data approach," in *proc. of IEEE COMPSAC (International Conference on Computers, Software & Applications)*, July 2018.

[39] A. P. Singh and G. J. Gordon, "A unified view of matrix factorization models," in *Machine Learning and Knowledge Discovery in Databases*, 2008, pp. 358–373.

[40] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[41] I. Netflix, "The netflix prize."

[42] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[43] X. Zhou, C. Yang, H. Zhao, and W. Yu, "Low-rank modeling and its applications in image analysis," *CoRR*, vol. abs/1401.3409, 2014. [Online]. Available: http://arxiv.org/abs/1401.3409

[44] S. Deerwester, S. T. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," vol. 41, pp. 391–407, 09 1990.

[45] B. Hutchinson, "Rank and sparsity in language processing," Ph.D. dissertation, University of Washington, 2013.

[46] G. Golub and C. Loan, *Matrix Computions*, 3rd ed. John Hopkins UP, 1996.

[47] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep 1936.

[48] M. Siderius and F. Jagodzinski, "Mutation sensitivity maps: Identifying residue substitutions that impact protein structure via a rigidity analysis in silico mutation approach," *Journal of Computational Biology*, vol. 25, no. 1, pp. 89–102, 2018.

[49] E. Andersson and F. Jagodzinski, "Promuteht: A high throughput compute pipeline for generating protein mutants in silico," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, ser. ACM-BCB '17. New York, NY, USA: ACM, 2017, pp. 655–660. [Online]. Available: http://doi.acm.org/10.1145/3107411.3116251

[50] G. G. Krivov, M. V. Shapovalov, and R. L. Dunbrack, "Improved prediction of protein side-chain conformations with scwrl4," *Proteins: Structure, Function, and Bioinformatics*, vol. 77, no. 4, pp. 778–795, 2009.

[51] M. J. Bower, F. E. Cohen, and R. L. J. Dunbrack, "Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool." *J Mol Biol*, vol. 267, no. 5, pp. 1268–1282, 1997.

[52] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, "Scalable molecular dynamics with NAMD," *Journal of computational chemistry*, vol. 26, no. 16, pp. 1781–1802, 2005.

[53] D. Jacobs, A. Rader, M. Thorpe, and L. Kuhn, "Protein flexibility predictions using graph theory," *Proteins*, vol. 44, pp. 150–165, 2001.

[54] D. Jacobs and M. Thorpe, "Generic rigidity percolation: the pebble game," *Physics Review Letters*, vol. 75, pp. 4051–4054, 1995.

[55] I. Tsang and I. Tsang, "Cluster size diversity, percolation, and complex systems," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 60, pp. 2684–2698, 1999.

[56] A. G. Ladurner and A. R. Fersht, "Glutamine, alanine or glycine repeats inserted into the loop of a protein have minimal effects on stability and folding rates1," *Journal of Molecular Biology*, vol. 273, no. 1, pp. 330 – 337, 1997.

[57] D. Jacobs and B. Hendrickson, "An algorithm for two-dimensional rigidity percolation: the pebble game," *Journal of Computational Physics*, vol. 137, pp. 346–365, 1997.

[58] N. Srebro and T. S. Jaakkola, "Weighted low-rank approximations," in *Proc. ICML*, 2003.

[59] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[60] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, 2011.