Low Rank Smoothed Sampling Methods for Identifying Impactful Pairwise Mutations

Nicholas Majeske
Western Washington U.
Bellingham, WA
Majeskn@wwu.edu

Filip Jagodzinski[*]
Western Washington U.
Bellingham, WA
Filip.Jagodzinski@wwu.edu

Brian Hutchinson[†]
Western Washington U.
Bellingham, WA
Brian.Hutchinson@wwu.edu

Tanzima Islam
Western Washington U.
Bellingham, WA
Tanzima.Islam@wwu.edu

ABSTRACT

Even a single amino acid substitution in a protein can be the cause of a debilitating disease. Experimentally studying the effects of all possible multiple mutations in a protein is infeasible since it requires a combinatorial number of mutants to be engineered and assessed. Computational methods for studying the impact of single amino acid substitutions do not scale to handling the number of mutants that are possible for two amino acid substitutions. We present an approach for reducing the amount of mutation samples that need to be used to predict the impact of pairwise amino acid substitutions. We evaluate the effectiveness of our method by generating exhaustive mutations in silico for 8 proteins with 2 amino acid substitutions, analyzing the mutants via rigidity analysis, and comparing the predictions from a sample of the mutants to that in the exhaustive dataset. We show it is possible to approximate the effect of the two amino acid substitutions using as little as 25% of the exhaustive mutations, which is further improved by imposing a low rank constraint.

CCS CONCEPTS

• **Computing methodologies → Factorization methods**; • **Applied computing → Bioinformatics**; *Molecular structural biology*;

KEYWORDS

protein; mutations; big data; sampling; low-rank

ACM Reference Format:
Nicholas Majeske, Filip Jagodzinski, Brian Hutchinson, and Tanzima Islam. 2018. Low Rank Smoothed Sampling Methods for Identifying Impactful Pairwise Mutations. In *ACM-BCB'18: 9th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, August 29-September 1, 2018, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3233547.3233714

1 INTRODUCTION

Inferring the effects of amino acid substitutions has a wide range of applications. Knowing the extent to which a mutation alters a protein's stability can aid in drug design studies for delivering pharmaceutical solutions for combating diseases caused by protein mutants [25].

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) [27]. Unfortunately, mutagenesis experiments on physical proteins 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 complement 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 [9], and relied on heuristic energy functions or database-derived potentials [11]. Other approaches are dependent on sufficiently large datasets of homologous proteins [4, 31]. Approaches based on machine learning (ML), a branch of artificial intelligence, have also been leveraged to infer the effects of mutations. Some use support vector machines (SVMs) [7, 15], while others utilize random forests and similar approaches [16]. Several of these ML methods achieve high accuracy rates (upwards of 80%) predicting 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 [12, 30], even though 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 gene which confers drug-associated resistance to protease inhibitors is 28 [26].

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 [2], 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

[*] Corresponding author
[†] Dr. Hutchinson has a joint appointment with the Computing and Analytics Division of Pacific Northwest National Laboratory, Richland, WA.

ACM-BCB'18, August 29-September 1, 2018, Washington, DC, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5794-4/18/08…$15.00
https://doi.org/10.1145/3233547.3233714

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).

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 two amino acid substitutions, and used it to generate an exhaustive set of possible mutants for each of 8 proteins. We perform a quick analysis of the flexibility of the *in silico* generated mutant and wild type structures using a 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 exhaustive data 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 counter-act 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.

2 RELATED WORK

The majority of computational approaches for inferring the effects of mutations reason about the impact of single amino acid substitutions. PoPMuSiC 2.1 [8] makes predictions about ΔΔG and generates a sequence optimality score. AutoMute [20] is a ML-based method that requires a large training set. CUPSAT [22] relies on energy potentials (atomic and torsional angles), and is dependent on a radial pair distribution function, whose calculation is time intensive. D-Mutant [32] constructs a residue-specific all-atom potential and requires the use of 1,011 actual protein structures with resolution <2Å . 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 [6]. STRUM [24] is a physics-based energy calculation approach that relies on multiple-threading template alignment. McCafferty [21] 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 screening-like analyses. MAESTRO and MAESTROweb [18] are machine learning based approaches for predicting $\Delta\Delta G$ values for mutations, but do not permit a screening of all possible multiple-mutation variants. DUET [23], although it predicts $\Delta\Delta G$, does not allow a user to perform a screen in which a subset of pairwise mutations are assessed. In our most recent work, we developed a compute pipeline for generating *in silico* all mutants with pairwise mutations [19], and generated an Allostery Impact Map to identify pairs of residues that cause a disruption to the protein's stability.

Low rank matrix factorization is at the heart of a wide range of data analysis techniques; for example, the popular principal component analysis dimensionality reduction technique. Low rank is often found in matrices describing interactions between two types of entities. One famous example is the *Netflix* movie recommendation problem [3], 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 *kinds of users* and *kinds of movies*, and the interactions between them explain much of the ratings in user-movie ratings.

In Section 3.3, we explore low rank structure in matrices describing the effects of pairwise mutations in proteins. We use a singular value decomposition to find a low rank approximation based on the well-known Eckart-Young-Mirksy theorem [10].

3 METHODS

Exhaustive mutation sets have been used in the past to explore and identify impactful amino acid substitutions via Allostery Impact Maps (AIMs) [19, 28]. However, generating all possible mutants with two amino acid substitutions can take several weeks – even months – of compute time. In this paper, we present a multi-phase compute pipeline (Fig 1) for evaluating several sampling methods to reduce the number of pairwise mutations required to make good predictions about the effects of pairwise mutations.

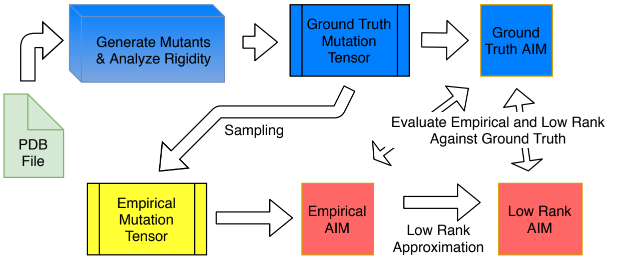
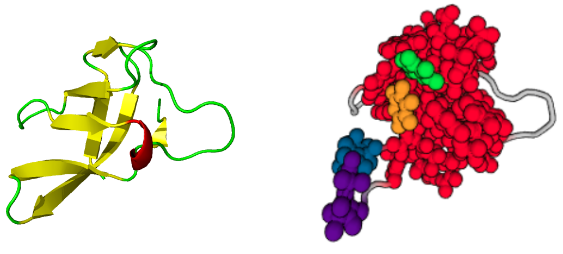

Generate Mutants & Analyze Rigidity
Ground Truth Mutation Tensor
Ground Truth AIM
PDB File
Sampling
Evaluate Empirical and Low Rank Against Ground Truth
Empirical Mutation Tensor
Empirical AIM
Low Rank Approximation
Low Rank AIM

Figure 1: Pipeline: We generate an exhaustive mutation set (blue), sample (yellow), and generate empirical and low rank Allostery Impact Maps (red).

Our pipeline improves the scalability of the prediction operation for larger proteins and resulting exhaustive pairwise mutation sets. We also evaluate the quality of the proposed sampling methods by quantifying how close the predictions are with respect to exhaustive AIMs. Our pipeline is comprised of three phases. ***Phase 1: Generating exhaustive pairwise mutation set*** – This phase is required for validating the effectiveness of our sampling methods. There are two tasks: (i) identifying the exhaustive set of all possible mutants having two amino acid substitutions, and (ii) analyzing the effects of pairwise mutations using rigidity analysis. ***Phase 2: Sampling from exhaustive pairwise mutation set*** – We apply our proposed sampling methods to specifically study the impact of pairwise mutations to hydrophobic, hydrophilic, and pairwise mutations sampled at random. ***Phase 3: Low Rank smoothing*** – To improve the approximation quality of the empirical (sampled) AIMs, we impose a low rank constraint, producing a low rank AIM.



(a) cartoon

(b) rigidity results

Figure 2: Rigidity analysis for PDB file 1csp (a) identifies atoms belonging to the same rigid clusters (b).

This low rank smoothing reduces noise and improves approximation quality. We explain each of these phases, as well as details of the tasks involved, in the following subsections.

3.1 Exhaustive Pairwise Datasets

Generating Mutant Structures: For this work, we use the ProMuteHT software [1], and generated all possible mutants with two amino acid substitutions for 8 proteins (Table 1).

Table 1: PDB files used, and mutants generated

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

Rigidity Analysis: Rigidity analysis [14] is a fast graph-based method that identifies rigid regions of biomolecules [17]. Atoms and their chemical interactions are used to construct a mechanical model and associated graph of a protein, whose analysis via a pebble game algorithm [13] identifies rigid clusters of atoms (Fig 2).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 [28]) :

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

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 an **Exhaustive Mutation Tensor**, $\mathcal{T}^{ex} \in \mathbb{R}^{n\times n\times 361}$. The $(i,j,k)^{th}$ element, $\mathcal{T}^{ex}_{ijk}$, 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.

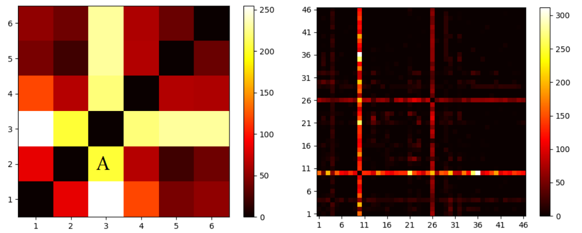
From $\mathcal{T}^{ex}$, we build an Exhaustive Allostery Impact Map (AIM), M^{ex}, [19] which provides an infographic (Fig 3) based on quantitative data for reasoning about the effects of mutating two residues.

Because of the large count of structures that make up an exhaustive pairwise mutation set for a protein, we distribute the computational tasks for Phase 1 among 165 machines. Each machine further subdivides each task via process-level parallelism by spawning 1 mutex process for mutation for each available compute core. We achieve a process-level granularity of $19^k\binom{n}{k}/(165\cdot 8)$ when generating all possible protein mutants containing $k = 2$ amino acid substitutions for a protein with n residues. Our compute pipeline leverages the knowledge that no two pairwise protein mutations depend on each other to parallelize the generation and analysis of all pairwise mutations, and the mutually independent computation tasks are run in a distributed computing environment.

3.2 Empirical Allostery Impact Map

In estimating the exhaustive AIM, we use three methods of sampling to derive three different empirical AIMs. These three methods include: sampling randomly from the set of all mutations ($\mathcal{T}^{emp}_{tr}$), sampling randomly from the set of mutations in which all substitutions are to hydrophobic amino acids ($\mathcal{T}^{emp}_{phob}$), and sampling randomly from the set of mutations in which all substitutions are to hydrophilic amino acids ($\mathcal{T}^{emp}_{phil}$).

For each of these three sampling methods, we analyze the quality of approximation as a function of the quantity of sampling. First, we sweep the number of mutation site pairs sampled in {25%, 50%, 75%, 100%} while holding the number of mutations sampled for each of these site pairs constant at 19. Additionally, in empirical AIMs for which less than 100% of mutation site pairs are sampled, we have ‘unfilled’ empirical AIMs where mutation site pairs are left unsampled, and ‘filled’ empirical AIMs 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%. In



(a) sample AIM

(b) AIM for pdb 1crn

Figure 3: Allostery Impact Maps : A cell color specifies the sum values for the rigidity metric for all 361 mutants generated by exhaustively mutating the amino acids indicated by 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 [19].

this phase, our pipeline again leverages a distributed computing environment; we distribute the exhaustive pairwise mutation sets for 8 proteins to 8 compute nodes for sampling.

3.3 Low Rank Allostery Impact Map

While our empirical AIMs are fast to generate, they paint an incomplete picture of the exhaustive AIM. "Filling in" the missing information requires making some assumption about global structure of the exhaustive AIM. We assume that the exhaustive AIM is *low rank*. The rank of a matrix is the number of linearly independent columns (and rows) in the matrix; equivalently, it is 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 exhaustive AIMs for the proteins we 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:

$$M^{emp}_R = U\Sigma_R V^T. \tag{4}$$

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^{emp}_R, 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 computational experiments and results show that the approach works well, and we leave weighted approximations [29] to future work.

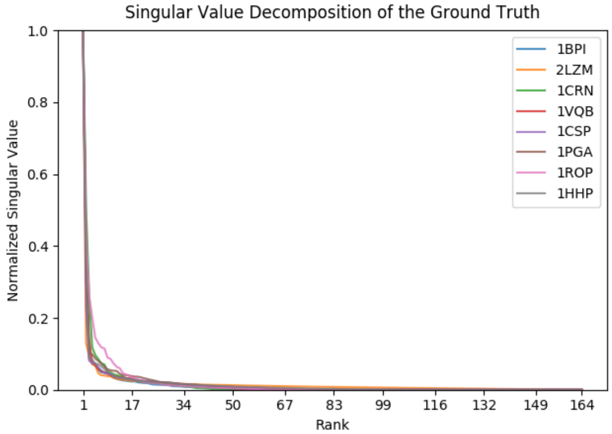


Figure 4: Singular values for 8 proteins, revealing approximate low rank structure. Singular values were normalized by dividing by the largest singular value.

3.4 Evaluation Metrics

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

$$SAE = \sum_{i=1}^{n}\sum_{j=1}^{n} |M^{gt}_{ij} - M_{ij}| \tag{5}$$

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^{ex} and its SAE approaches zero.

4 RESULTS - CASE STUDIES

We evaluate the efficacy of our low rank smoothed sampling methods by computing the SAE compared to exhaustive over empirical approximation (the lower the better), and measuring how accurately a significantly reduced subset of the exhaustive mutation set can reconstruct the characteristic bands representing mutation sensitive sites.

4.1 Low Rank Versus Random Sampling

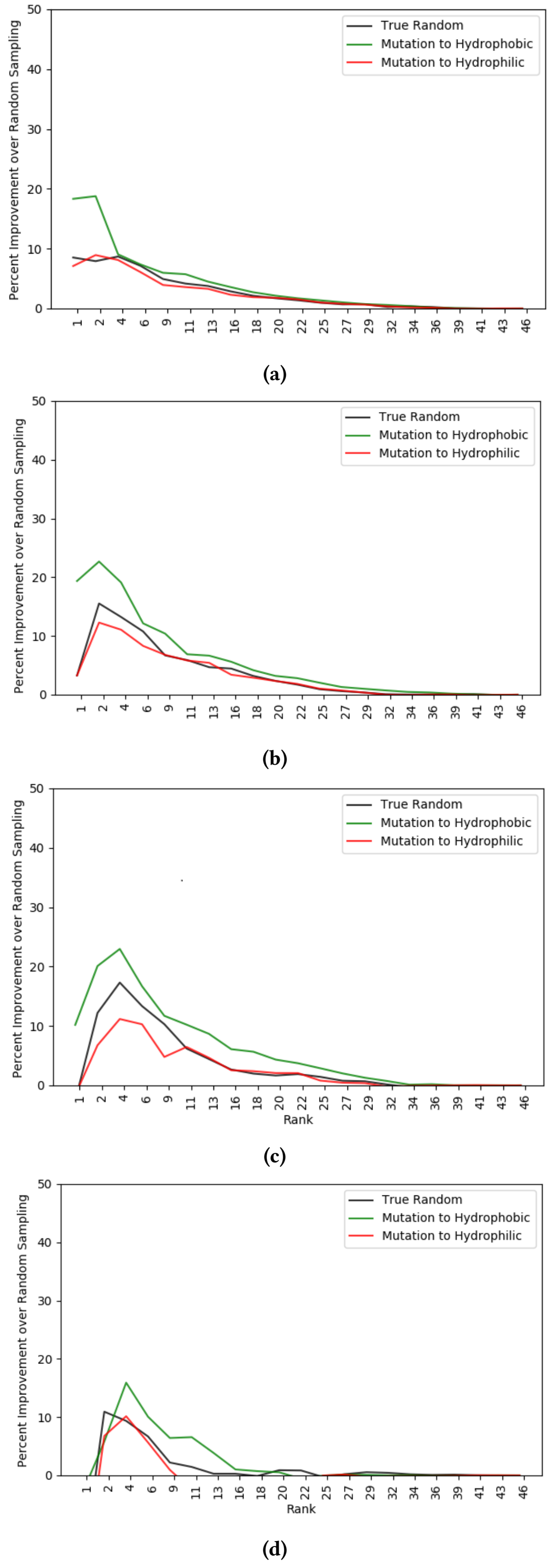
The low rank model consistently reduces SAE relative to the empirical model on 1crn for small values of the rank, R (Fig 5). As R approaches 46, M^{lr} approaches M^{emp} and the improvement converges to 0. The biggest improvements by smoothing are in the "Mutation to Hydrophobic" case, suggesting this subset of the data is particularly well-suited to the low rank assumption. Unlike 1crn, for 1pga (Fig 6) we see a distinctive increase in improvement as the fraction of sites sampled approaches 1.0, achieving a relative reduction in SAE of up to 34%. This indicates that all mutation sites encode unique information as opposed to 1crn where most information is encoded in a small number of sites.

4.2 Low Rank Approx. & Sampling Error

Fig 7 plots the absolute error (SAE) for the empirical and low rank models for the three sampling types (Y axis) across ranks (X axis), for the "unfilled" sampling strategy on 2lzm. There is a clear basin of good values of R ranging from 8-32. For sampling at 75% of pair-wise mutation sites for 1hhp (Fig 8), the"sweet spot" for the low rank approximation is at a much lower rank than for 2lzm. 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 might cause a surface residue to be energetically unfavorable.

4.3 Heatmaps

For 1crn (Fig 9, showing M^{ex}, M^{emp} and M^{lr}), the low rank model is able to detect the banded, low rank structure of the exhaustive AIM from the samples in the empirical data, but the heatmap tends to over-generalize. For other proteins (not shown), the low rank models achieve similar results.



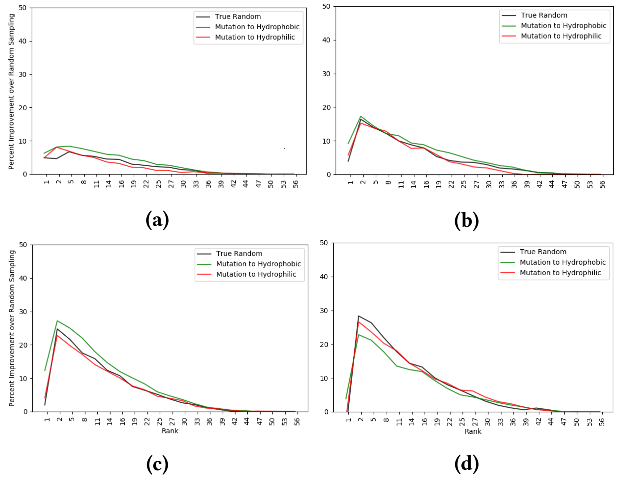
(a)

(b)

(c)

(d)

Figure 5: Improvement in SAE by low rank smoothing relative to "filled" empirical approx for 1crn. Randomly sampling 19 mutations across mutation site pairs at 25 (a), 50 (b), 75 (c) and 100% (d) sampling.



(a)

(b)

(c)

(d)

Figure 6: Improvement in SAE by low rank smoothing relative to "filled" empirical approx for 1pga. Randomly sampling 19 mutations across mutation site pairs at 25(a), 50(b), 75(c) and 100%(d) sampling.

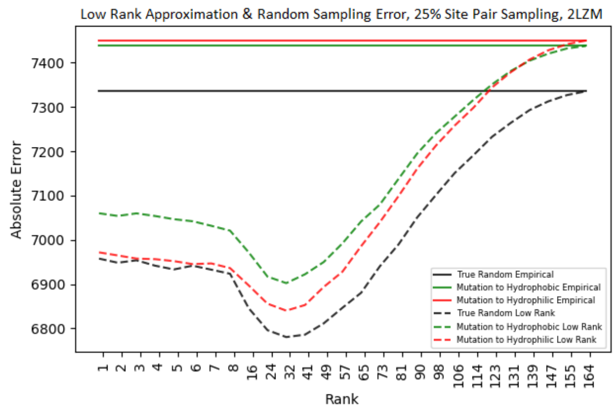


Figure 7: Empirical approximation error against low rank approximation error for various ranks when sampling across 25% of mutation site pairs for 2lzm.

5 CONCLUSIONS & FUTURE WORK

We exhaustively generated mutant sets for 8 proteins, and analyzed both the wild type and mutants using rigidity analysis; we call this data our ground truth. Because even computational approaches for such exhaustive screens are time consuming, we have presented several methods to accurately approximate the exhaustive data using a fraction of the total samples from the exhaustive set.

We observed several interesting results when comparing the exhaustive, empirical approximation, and low rank approximation Allostery Impact Maps. In some proteins – 2lzm – 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% from the exhaustive mutation set. For some proteins – 1pga – many mutation sites encode unique information, but for others such as 1crn, most information about the effects of pairs

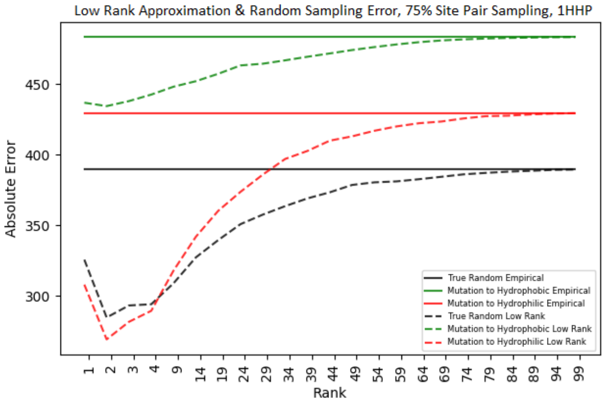


Figure 8: Empirical approximation error against low rank approximation error for various ranks when sampling across 75% of mutation site pairs for 1hhp.

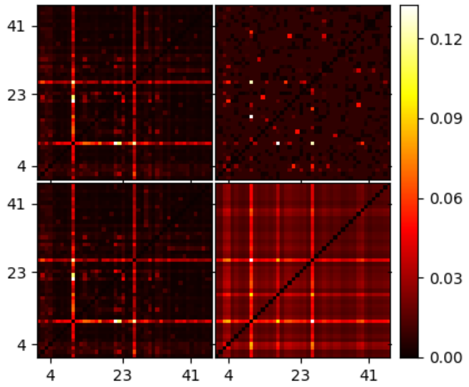


Figure 9: 1crn : Exhaustive (left), empirical approximation (upper right) and low rank approximation (low right) AIMs.

of substitutions was encoded in a small number of sites. The fact that a choice of 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.

There are ways the low rank approximation in this work could be extended. For example, weighted low-rank decompositions [29] would likely improve the quality of the low rank approximation. It would also be worth exploring low rank decompositions explicitly designed to be robust to noise [5].

REFERENCES

[1] E Andersson and F Jagodzinski. 2017. 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 (ACM-BCB '17)*. 655–660.

[2] A Bava, M Gromiha, H Uedaira, K Kitajima, and A Sarai. 2004. ProTherm, v 4.0: thermodynamic database for proteins and mutants. *Nucleic acids res* 32, suppl 1 (2004), D120–21.

[3] J Bennett and S Lanning. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, Vol. 2007. New York, NY, USA, 35.

[4] J Brender and Y Zhang. 2015. Predicting the effect of mutations on protein-protein binding interactions through structure-based interface profiles. *PLoS Comp Bio* 11, 10 (2015).

[5] E. Candès, X Li, Y Ma, and J Wright. 2011. Robust Principal Component Analysis? *J. ACM* 58, 3 (2011), 11:1–11:37.

[6] E Capriotti, P Fariselli, and R Casadio. 2005. I-Mutant2 : predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Res.* 33, suppl 2 (2005), W306–W310.

[7] J. Cheng, A. Randall, and P. Baldi. 2006. Prediction of Protein Stability Changes for Single-Site Mutations Using Support Vector Machines. *PROTEINS: Struct Func & Bioinfo* 62 (2006), 1125–1132.

[8] Y Dehouck, J Kwasigroch, M Gilis, and Rooman M. 2011. PoPMuSic2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. *BMC Bioinfo* 12 (2011).

[9] R Dunbrack and M. Karplus. 1994. Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains. *Nature Structural Biology* 1 (1994), 334–340. Issue 5.

[10] C Eckart and G Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika* 1, 3 (01 Sep 1936), 211–218.

[11] D. Gilis and M. Rooman. 1997. 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* 272, 2 (1997), 276–290.

[12] S Henikoff and PC Ng. 2006. Predicting the effects of amnio acid substitutions on protein functions. *Annual Reviews of Genomics Human Genetics* 7 (2006), 61–80.

[13] D Jacobs and B Hendrickson. 1997. An algorithm for two-dimensional rigidity percolation: the pebble game. *J. Comput. Phys.* 137 (1997), 346–365.

[14] D Jacobs, AJ Rader, M Thorpe, and L Kuhn. 2001. Protein Flexibility Predictions Using Graph Theory. *Proteins* 44 (2001), 150–165.

[15] F Jagodzinski, B Akbal-Delibas, and N Haspel. 2013. An Evolutionary Conservation & Rigidity Analysis Machine Learning Approach for Detecting Critical Protein Residues. In *Proc. of the Int'l Conf on Bioinformatics, Computational Biology and Biomedical Informatics*. ACM, New York, NY, USA, 779–785.

[16] L Jia, R Yarlagadda, and C Reed. 2015. Structure Based Thermostability Prediction Models for Protein Single Point Mutations with Machine Learning Tools. *PloS one* 10, 9 (2015), e0138022.

[17] Andreas G. Ladurner and Alan R. Fersht. 1997. Glutamine, alanine or glycine repeats inserted into the loop of a protein have minimal effects on stability and folding rates1. *Journal of Molecular Biology* 273, 1 (1997), 330 – 337.

[18] J Laimer, H Hofer, M Fritz, S Wegenkittl, and P Lackner. 2015. MAESTRO-multi agent stability prediction upon point mutations. *BMC bioinformatics* 16, 1 (2015), 116.

[19] N Majeske and F Jagodzinski. 2018. Elucidating Which Pairwise Mutations Affect Protein Stability: An Exhaustive Big Data Approach. In *proc. of IEEE COMPSAC (International Conference on Computers, Software & Applications).*

[20] M Masso and I Vaisman. 2010. AUTO-MUTE: web-based tools for predicting stability changes in proteins due to single amino acid replacements. *Protein Engineering Design and Selection* 23, 8 (2010), 683–687.

[21] C McCafferty and Y Sergeev. 2016. In silico Mapping of Protein Unfolding Mutations for Inherited Disease. *Scientific Reports* 6 (2016), 37298.

[22] V Parthiban, M Gromiha, and D Schomburg. 2006. CUPSAT: prediction of protein stability upon point mutations. *Nucleic Acids Res* 34, suppl 2 (2006), W239–W242.

[23] D Pires, D Ascher, and T Blundell. 2014. DUET: a server for predicting effects of mutations on protein stability using an integrated computational approach. *Nucleic acids research* 42, W1 (2014), W314–W319.

[24] L Quan, Q Lv, and Y Zhang. 2016. STRUM: structure-based prediction of protein stability changes upon single-point mutation. *Bioinfo* 32, 19 (2016), 2936–2946.

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

[26] SY Rhee, J Taylor, J Fessel, D Kaufman, W Towner, P Troia, P Ruane, J Hellinger, V Shirvani, A Zolopa, and R Shafer. 2010. HIV-1 Protease Mutations and Protease Inhibitor Cross-Resistance. *Antimicrobial Agents & Chem.* 59, 8 (2010), 4253–4261.

[27] J Schellman. 1987. The thermodynamic stability of proteins. *Annual rev. of biophysics and chem* 16, 1 (1987), 115–137.

[28] M Siderius and F Jagodzinski. 2018. Mutation Sensitivity Maps: Identifying Residue Substitutions That Impact Protein Structure Via a Rigidity Analysis In Silico Mutation Approach. *J of Comp Bio* 25, 1 (2018), 89–102.

[29] N Srebro and TS Jaakkola. 2003. Weighted low-rank approximations. In *Proc. ICML*.

[30] S Teng, E Michonova-Alexova, and E Alexov. 2008. Approaches and resources for prediction of the effects of non-synonymous single nucleotide polymorphisms on protein function and interactions. *Cur. Pharmacology Biotech.* 9 (2008), 123–133.

[31] CL Worth, R Preissner, and L Blundell. 2011. SDM-a server for predicting effects of mutations on protein stability and malfunction. *Nucleic Acids Research* 39, Web Server Issue (2011), W215–W222.

[32] H Zhou and Y Zhou. 2002. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein science* 11, 11 (2002), 2714–2726.