Assessing the Effects of Amino Acid Insertion and Deletion Mutations

Muneeba Jilani
Dept. of Computer Science
Univ. of Massachusetts Boston
Boston, MA, USA
muneeba.jilani001@umb.edu

Alistair Turcan
Dept. of Computer Science
Western Washington Univ.
Bellingham, WA USA
turcana@wwu.edu

Nurit Haspel
Dept. of Computer Science
Univ. of Massachusetts Boston
Boston, USA
nurit.haspel@umb.edu

Filip Jagodzinski
Dept. of Computer Science
Western Washington Univ.
Bellingham, WA USA
filip.jagodzinski@wwu.edu

2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-0126-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIBM52615.2021.9669484

Abstract

Despite being a recurring type of sequence variation, amino acid insertions and deletions (InDels) and their resulting functional significance remain a rather unexplored area of structural biology. InDels are quite often the driving force behind many diseases. Despite that, modeling InDels and exploring their functional implications remains lacking, mainly due to the dearth of experimental information and computational methodologies. In this work we introduce an algorithmic approach to model short InDels *in silico* and explore the structural and rigidity differences between the wildtype and the mutant protein structures. We assess rigidity to gather useful information about the protein's general structure, the location of flexible and rigid clusters, and to permit a visual analysis of the effects of InDels local to the mutation site. Our results show that our method can efficiently create a computer-generated mutant that is functionally similar to the experimental mutant at both the local region of the InDel, as well as on the entire protein scale. The results show promise in our ability to accurately predict the effects of short insertions and deletions on the structural properties of proteins.

Keywords:: computational structural biology, protein InDel mutations, graph-theory rigidity

I. Introduction

Amino acid insertions and deletions (InDels) are a common type of sequence variation in proteins. While the mechanisms and functional changes caused by amino acid substitutions have been studied extensively, InDels remain less understood and studied due to the challenges of conducting wet-lab experiments in which an inDel at the sequence level is followed by transcription and translation that results in a viable protein structure [1]. For this reason, the effects of InDels on protein structure and dynamics is less studied [2]. InDel events occur when a non-frameshift (NFS) insertion or deletion in the DNA sequence results in one or more amino acids being inserted or deleted relative to the wildtype amino acid sequence in a protein. A protein variant that includes one or more inserted or deleted amino acids relative to the wild type is called an InDel protein mutant [3, 1, 4]. DNA inDels that cause protein InDels have various causes such as genome duplication, proliferation of transposal elements, as well as replication errors [3]. While InDels can occur anywhere in a protein sequence, they are observed more often in the loop regions of proteins [5], possibly due to their disruptive consequences if they occur otherwise in secondary structure elements [6]. Further, InDels usually do not occur in loop regions that are involved in a catalytic reaction or a triad, or which serve a structural role. It has recently been demonstrated that InDels, rather than substitutions, are strongly correlated with functional changes in proteins [7]. In sum, there is substantial evidence that InDels, not substitutions, are a predominant evolutionary factors when it comes to structural changes in proteins [8, 9, 10].

InDels can result in many types of diseases such as cystic fibrosis [11] and several types of cancer [12, 13]. The more recently occurring severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [14] has variants that are caused by InDels. A closer look at the spike protein of this virus reveals that InDels which occur at the S1/S2 subunits results in mutants that have greater probability of resistance to vaccines. In the case of cystic fibrosis, several mutants are known. The F508del mutation in nucleotide-binding domain-1 (NBD1) of the cystic fibrosis transmembrane conductance regulator (CFTR) is the predominant cause of cystic fibrosis [11].

A. Previous Work

Early work at cataloging proteins with InDels included Indel PDB [15], which is no longer publicly available. It was a database of InDels generated from sequence alignments of highly similar proteins. A later tool was Sequence Feature and InDel Region Extractor (SeqFIRE) [16], which automated identification and extraction of InDels from protein sequence alignments. SeqFIRE extracted conserved blocks and identified fast evolving sites using a combination of conservation and entropy information about InDel locations. Like InDel PDB, SeqFIRE is no longer publicly available. Not many tools exist today that model the structural changes in a protein in response to an InDel, nor are there any tools that attempt to correlate structural changes due to InDels and their ensuing stability of a protein.

In this work we develop a computational pipeline, employing inverse kinematics and rigidity analysis, to model short InDels in protein structures. Our goal here is twofold: First, we aim to validate our protocol using wildtype-InDel pairs from experimentally resolved PDB structures, and compare them to the computationally generated InDels. We also investigate the structure and stability properties of the *in silico* generated InDels and their wildtype equivalents, to explore possible mechanisms to assess the extent to which an InDel affects a protein's structure and function.

II. Methods

Our methodology includes identifying wildtype and their InDel mutants in the PDB [17], generating the *in silico* InDel mutants via an inverse kinematics robotics inspired approach, and comparing the PDB mutants to our *in silico* generated mutants via a rigidity analysis approach.

A. Identifying InDels in the PDB

Because there are no publicly-available well-established databases of experimentally resolved InDel mutant protein structures, we proceeded with a multi-pronged approach to identify our data set of wildtype, InDel mutant pairs. One source of such wildtype, InDel pairs was UniProt [18], from which we retrieved PDB codes of wildtype and corresponding InDel mutant structures. We also conducted an advanced search of the PDB, looking for keywords such as *deletion*, *insertion* and *mutation* in the title. This yielded structures such as 6IDC [19]. Lastly, we searched the SEQADV and 999 REMARKS records among the entries in the PDB for *insertion* and *deletion* codes, which allude to InDels.

All of the found InDel mutants were further manually curated to determine the count of InDels, as well as their locations – whether in a secondary structure or a loop region. In total, we identified 24 mutants with PDB structures with InDels in a loop region. For this We restricted our search to InDels of size 1-4. With the help of UniProtKB [20], we identified their corresponding wildtype structures.

B. Creating InDels and refining the resulting structures

The *in silico* InDels were generated from the wild type PDB structures using the Rosetta software suite [21]. Rosetta is a widely used molecular modeling application, which offers a wide range of tools for modeling and performing three-dimensional structure predictions and designing novel protein-protein complexes. Rosetta's protocols were used for loop modeling applications and *ab initio* energy minimization.

In recent years, geometry and robotics based algorithms have been used to model protein structure and dynamics [22, 23]. Robotics-inspired motion planning methods model the protein as a kinematic chain of rigid bodies connected by flexible joints [24]. The motion of proteins is simulated through the manipulation of degrees of freedom (DoF) of the protein bond lengths, bond angles, and dihedral angles. Inverse Kinematics (IK) is used to model the configuration of a kinematic chain given its end constraints [23]. In protein structure prediction, IK is often used to model loops by manipulating the rotational degrees of freedom of a loop region to find possible loop conformations that attach to the rest of the protein. Inspired by IK, KIC (Figure 1)[25] is a robotics inspired methodology used for computation of probable arrangements of linked objects that are subjected to a set of constraints. Initially the loop is divided into three pivots (colored green in the figure) and the rigid body transformation from the start of the loop till the end is stored (highlighted by black dotted line). This is followed by perturbing the degrees of freedom [25]. Lastly, pivot torsion values are calculated and set to orient each rigid segment so that the original rigid body transformation between the loop ends is preserved. [26].

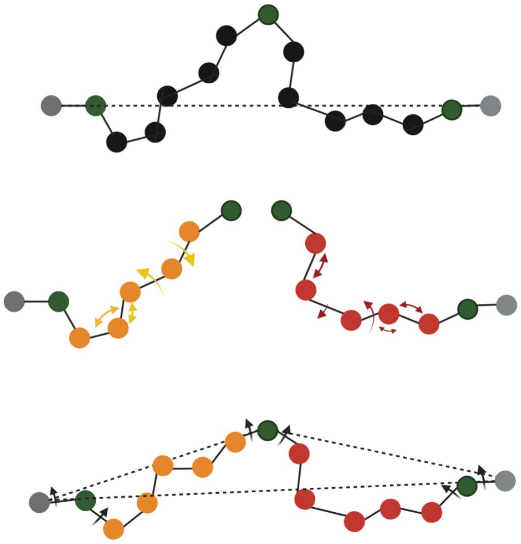


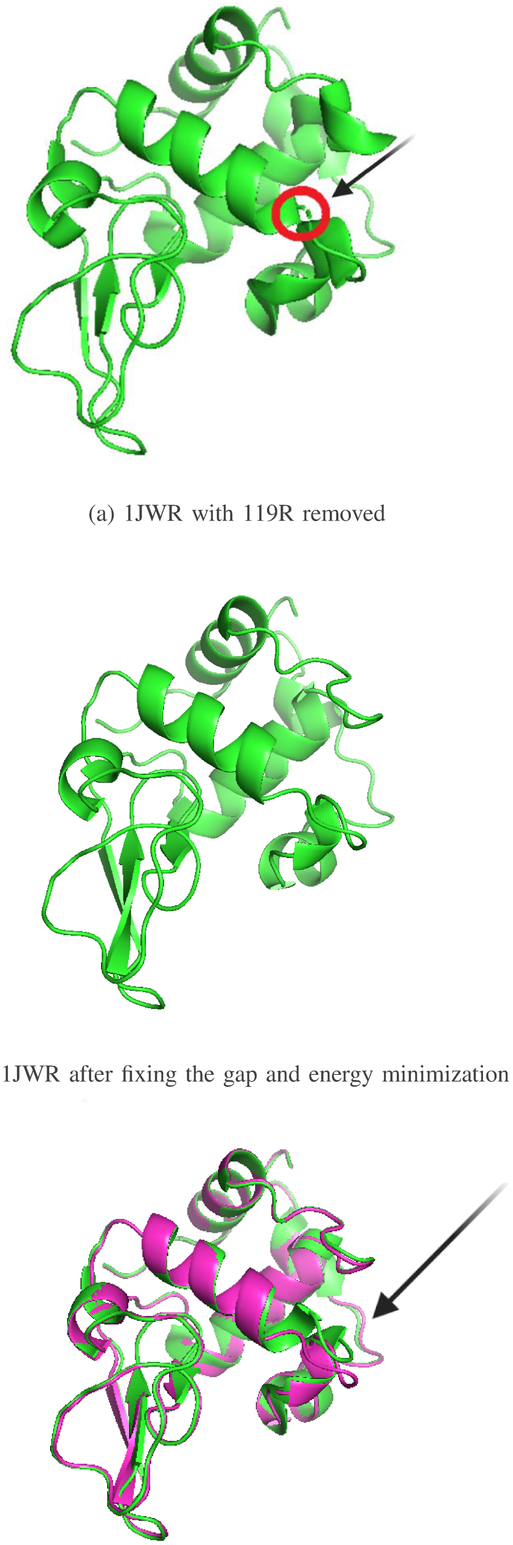
Fig. 1: A demonstration of kinematic closure, (top to bottom) the first figure shows that rigid body transformations are computed and stored shown by black dotted line. Second figure shows perturbing the degrees of freedom pertaining torsions, bond lengths and bond angles depicted by orange and red arrows and the third figure shows that pivot torsion values are calculated and stored shown by black arrows and each rigid segment is oriented in a manner to reinstate the original rigid body transformation [25]

For a kinematic articulated chain with a gap, that represents an amino acid deletion or location where an amino acid is inserted, the problem consists of finding a balance between global tweaking of the torsion angles, and making positional and angle changes incident to the residues adjacent to the deleted or inserted residue. Our algorithmic approach to address this challenge consists of two steps :

- Loop closure without refinement of resulting structures using geometric modeling.
- Perform KIC in order to resolve the loop for energy minimization.

The specific protocols used to achieve this are as follows: Rosetta's *remodel* protocol [27] was used to close the gap initially, followed by the *loop* protocol [28, 29]. The structures produced by Rosetta's *quick and dirty* variant of the *remodel* protocol do not have good energy scores as it simply closes the gap but the resulting structures have steric clashes. The *loop* protocol uses KIC as explained earlier to refine the region of closure, followed by all-atom refinement in order to find a low energy conformation. The last step results in better energy scores for the emergent structures.

For validation purposes, we compared the *in silico* generated mutant structures with their PDB counterparts and evaluated them based on both local (loop region) and global RMSD before passing them on to the rigidity analysis step of our approach. Figure 2 illustrates the process of deleting ARG-119 for human lysozyme [30], followed by closing the gap. Figure 2 (c) shows a superimposition of the computationally generated mutant with the experimentally available structure (PDB:1DI5). The global all atom RMSD was 0.283Å. A black arrow points to the region where the computational and experimental mutant differ the most. We also computed the local RMSD of the loop region from which the residue was deleted. In this case the local RMSD is 0.251Å.



(a) 1JWR with 119R removed

(b) 1JWR after fixing the gap and energy minimization

(c) 1DI5 computational mutant superimposed with 1DI5 with global RMSD of 0.283

Fig. 2: The loop closure process

Rigidity Analysis

The rigidity and flexibility of protein domains provides insights into a protein's structural stability [31]. A computational analysis of the rigidity of a protein can yield insights into the effects of amino acid substitutions [32, 33]. We relied on the software tool KINARI [34] to assess the rigidity of our InDel mutants. KINARI takes a PDB file as input, identifies stabilizing interactions such as hydrogen bonds, models the protein as a Body-Bar-Hinge Framework, and runs a Pebble Game analysis on an associated graph representing the Body-Bar-Hinge Framework. The output for rigidity analysis is a list of atoms that exist among the identified rigid clusters.

We measured the differences between the *in silico* generated InDel mutants and their wild types, and between the PDB InDel mutants and the wildtype, using RMSD measurements, and via a visual inspection via PyMol. The similarities and differences in these scores gives us insight into the ability of our method to accurately generate *in silico* InDels.

C. Rigidity Analysis to Measure InDel Effect

To quantify the difference between a wildtype and an InDel mutant, we used our previously developed Rigidity Distance Similarity Metric (RDSM), which reasons about the counts and sizes of rigid clusters in the wildtype versus a mutant [35]. The following RDSM score:

$$RDSM = \sum_{i=1}^{x=LRC} i \times w(x) \times [WT_i - Mut_i]$$

quantifies the cumulative differences between the count of the various sizes of rigid clusters (*i* is the size of the rigid cluster, and *LRC* = Largest Rigid Cluster) in the wild type (WT) and mutant (Mut). The sigmoid-based *w(x)* function

$$w(x) = \frac{1}{1 + e^{-0.1x+5}}$$

weighs the differences that exist for the larger clusters in the mutant and wild type more heavily than differences that

TABLE I: Global and local RMSD of PDB and *in silico* generated InDel mutants in Angstroms

Wildtype	Mutant	InDel type	Global RMSD	Local RMSD
5GQL	5GQI	d:1	0.138	0.126
5GQL	5GQJ	d:2	0.767	0.708
5GQM	5GQN	d:3	0.836	0.678
6J6C	6AIS	d:2	0.96	0.509
6J6C	6ICS	d:4	0.135	0.34
2VJI	4XQF	d:2	0.16	0.135
2BBO	1XMJ	d:1	0.652	0.768
2IQ1	6AK7	d:3	0.79	0.392
2VJJ	6GVP	d:2	0.186	0.376
1A7N	1A7O	d:1	0.181	0.41
1STN	1STA	i:2	0.506	0.47
2VJJ	6GVR	d:2	0.168	0.329
5GQM	5GQK	d:3	0.796	0.269
1JWR	1DI4	d:2	0.563	0.498
1JWR	1DI5	d:1	0.283	0.251
2Y0G	4KA9	i:1, d:1	0.72	0.58
4KJK	4KJL	i:1	0.378	0.574
2NIP	1RW4	d:1	0.557	0.53
4EUL	6FLL	d:2	0.206	0.36
1ANF	1MDQ	i:1	0.701	0.607
1OMF	1GFN	d:6	0.245	0.342
1F21	1GOA	i:1	0.97	0.85
2Y0G	4KAG	i:1, d:1	0.69	0.74
5YHA	5YHB	d:3	0.623	0.807

exist among smaller clusters, as differences among large rigid clusters are more important to a protein's rigidity.

We calculated the pairwise RDSM scores for the wildtype, PDB InDel mutant, and wildtype, *in silico* InDel mutant, to assess the quality of our computer approach for generating and assessing the effectgs of the InDels. The lower the RDSM score, the closer the rigidity properties of the wildtype and mutant are to each other. Since proteins vary in size, we normalized the RDSM scores by the size of the protein in order to better visualize the differences as a percent difference, and not an absolute difference.

We also performed a visual analysis of the rigid clusters at the location of the InDel using a custom built PyThon visualizer. We did this to glean insights about the local effects of the insertion or deletion in addition to the more global-based measurement provided by the RDSM metric.

III. Results and Discussion

A. Comparing PDB and Computed InDel Mutants

Table I shows the global RMSD and local (to the InDel region) RMSD between the PDB InDel mutant and our *in silico* generated InDel mutant. Some of the entries refer to varying-length InDels from the same protein. For instance, 6AIS and 6ICS represent deletions of two and four amino acids, respectively, from the loop region of outer surface protein A of *Borreliella burgdorferi*. In all, the table represents nineteen examples of deletions, and five insertions. The Maximum number of inserted residues is two. We were able to insert up to four residues in the loop region of the human lysozyme, but we were unable to find any preexisting InDel mutants in the PDB that contained more than two insertions in the loop region of that protein. One of the deletions is of length 6.

As can be seen, both the global and local RMSD values, in Angstroms, are all less than 1.0Å, with several as low as 0.13Å. Therefore our *in silico* approach for creating InDel protein mutants using the robotics-inspired inverse kinematics approach appears to yield structures that are similar to InDel mutants whose structures are resolved experimentally.

B. Using Rigidity Analysis to Measure the effect of an InDel

For this work, we rely on our RDSM scores for the InDels and wildtype to determine the extent that the indel in the PDB mutant had the same effect on the wildtype as the indel in the *in silico* generated mutant. In our previous work [35] we found that two RDSM metrics, which differ only by their $w(x)$ functions, produced RDSM scores that correlated best with the known effects of substitutions:

$$RDSM2 : w(x) = \frac{1}{1 + e^{-0.1x+5}}$$

$$RDSM3 : w(x) = \frac{1}{1 + e^{-0.05x+5}}$$

The RDSM2 and RDSM3 scores for the wildtype and PDB and *in silico* generate mutants among our dataset of protein structures are shown in Figures 3 and 4. These represent the differences in rigidity properties between the wildtype and InDel pairs. If the PDB InDel and our *in silico* generated InDel have the same effect on the rigidity properties of the protein, we can expect the RDSM between the wildtype and PDB InDel (blue bar) and the wildtype and *in silico* InDel mutant (orange bar) to be similar, in Figures 3 and 4. If our computational modelling of the InDel is correct, we can expect that the rigidity scores between the PDB and *in silico* InDel mutant, represented by the gray bars, to be small. In other words, we would expect the score between the wildtype and PDB mutant to be similar to the score between the wildtype and *in silico* generated mutant because that means that both mutants show a similar amount of difference from the wildtype. We also would expect the rigidity properties of the experimental and computational InDel mutants to be similar, providing evidence that we are able to successfully computationally model the Indel mutant.

C. Statistical Validation

To further determine whether using the RDSM metric to assess the effect of an InDel is a fair approach, we performed a statistical analysis of the results. The problem at hand can be divided into two parts: Let us denote the RDSM score between the PDB mutant and wildtype by X, and the RDSM score

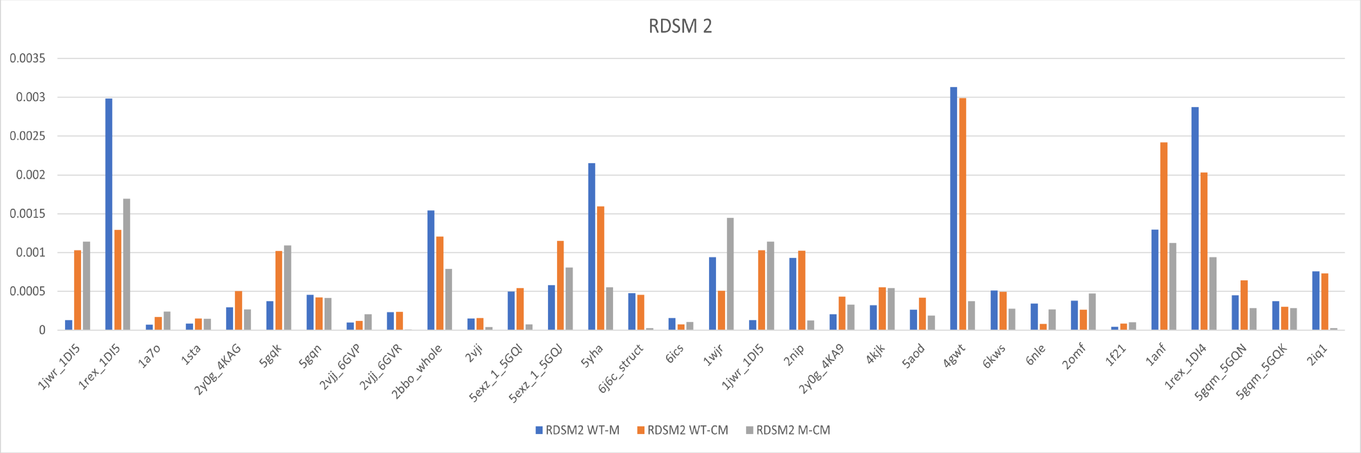


Fig. 3: The RDSM2 results for our tested proteins. The orange bar is a comparison of the RDSM2 measure between the wildtype and PDB InDel mutant. Blue a comparison of the RDSM2 between the wildtype and our Computer-Generated InDel Mutant, and the Gray bar is a comparison of the RDSM2 between the Mutant and Computer-Generated Mutant

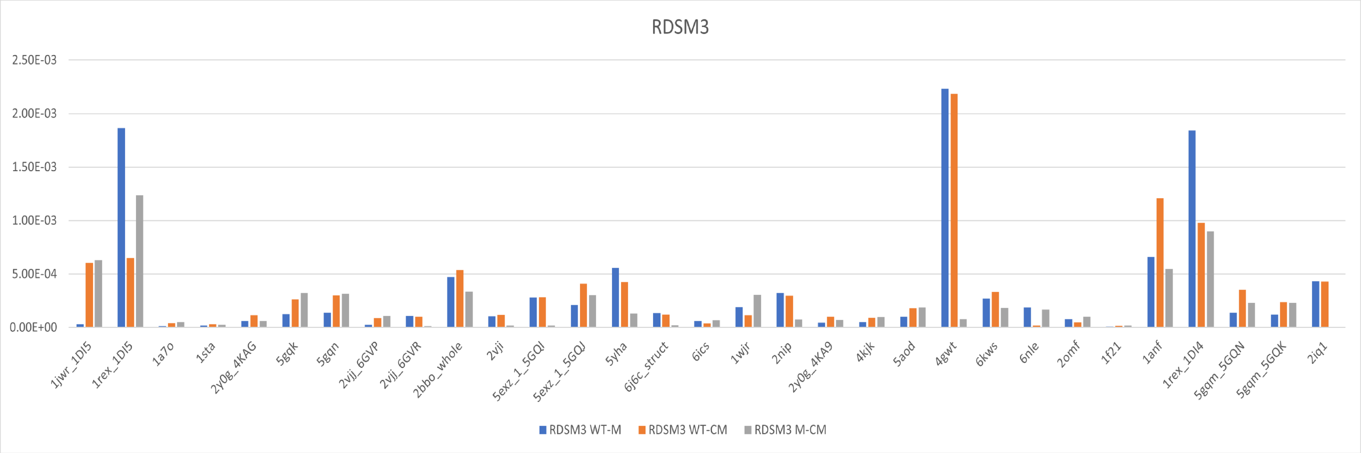


Fig. 4: The RDSM3 results for our tested proteins. The orange bar is a comparison of the RDSM3 measure between the wildtype and PDB InDel mutant. Blue a comparison of the RDSM3 between the wildtype and our Computer-Generated Mutant, and the Gray bar is a comparison of the RDSM3 between the Mutant and Computer-Generated Mutant

between the *in silico* mutant and wildtype by Y (representing the blue and orange bars in the graphs, respectively). To see whether they are similar we performed a Spearman correlation test [36] between the two variables. We conducted the test for 5 RDSM values (see [35] for the weights RDSM1 and RSDM 4-5), which differed only by their weight, $w(x)$ function. For RDSM1 through 5, we got values 0.93, 0.916, 0.897, 0.95 and 0.45 with statistical significance of $p < 0.05$. This statistical test revealed that both PDB mutant vs. wildtype and our *in silico* mutants vs. wildtype exhibited a satisfactory correlation to one another.

The second part of the problem consisted of finding what is the relation between X and Y above (blue and orange bar in the Figures 3 and 4, respectively) with the RDSM score between the PDB mutant and *in silico* mutant denoted by W (Grey bar in the graphs). The problem at hand now consisted of comparing W with X and Y. This posed a problem, since we have three magnitudes involved, and in order to compare their means we could not use a simple 2-sample $z-$ or t-test. Therefore, we relied on the following strategy to solve the problem at hand:

We computed the probability of the value of the third variable being smaller than the minimum of other two i.e. $P(W < min(X, Y)$. In order to achieve this, we computed the difference between W and the minimum of X and Y. Once we calculated the difference, which we denote as D, we computed the probability that $D < 0$. Any probability greater than 0.5 indicates that probability is within reasonable limits. We proceeded to repeat the same test for the RDSM values and attained the scores of 0.7638, 0.6533, 0.6823, 0.7098 and 0.6098 for RDSM 1-5. This indicated that the PDB and *in silico* InDel mutants are similar to one another as compared to the wildtype when analysis was performed on the current data.

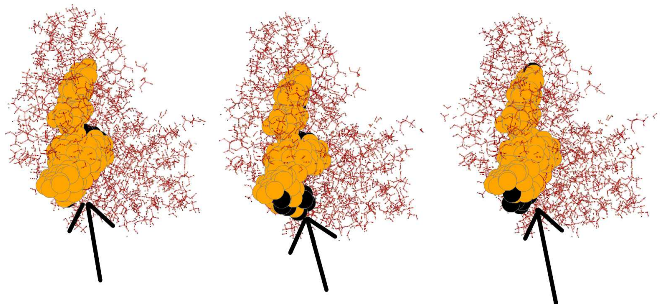


Fig. 5: Visualization (Left-to-right) of the wildtype 1A7O, experimental (PDB) mutant 1A7N, and computer generated mutant. The InDel was a deletion of Proline at residue 95. Residues surrounding the InDel location are made larger with an arrow drawn to the indel's exact spot. The largest 5 rigid clusters are colored orange, with all other clusters colored black

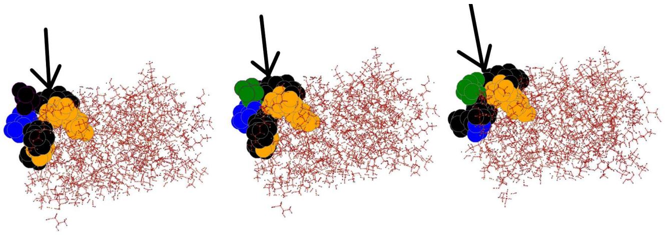


Fig. 6: Visualization (left-to-right) of wildtype 2Y0G, its experimental (PDB) mutant 4KAG, and computer generated mutant. The indel was an insertion of Aspartic Acid at residue 190. Residues surrounding indel location are made larger with an arrow drawn to the indel's exact spot. Largest 5 rigid clusters are colored, with all other clusters colored black

D. *Effects of InDels Local to Mutation Site*

We also assessed the effects of the InDel local to where the insertion or deletion were made, via a visual inspection using a custom built Python script. Figure 6 shows that despite the fact that the RDSM score comparison reveals that for the wildtype 2Y0G and PDB mutant 4KAG, our *in silico* generated InDel had a 2x greater effect than the PDB InDel on the RDSM, the visualization local to the InDel site is similar for the PDB mutant to the *in silico* mutant. Another example that showcase how the rigidity analysis approach to determine the effect of the InDel local to the mutation is shown in figure 5 that uses wildtype 1A7O and PDB mutant 1A7N. While this mutation didn't have as large of an effect on the wildtype, our *in silico* approach still resulted in rigid clusters that were very similar to the rigid clusters as found in the PDB InDel mutant.

IV. Conclusion

InDels account for more changes in the structure and function of proteins when compared to substitutions. However, they are not as well studied, among other reasons due to lack of experimental and computational data. In this research we attempted to gain insight related to InDels and their structural implications. The goal of this work was to computationally generate short insertions and deletions from PDB files and to predict the effects of the mutations. For the purpose of this research, we identified twenty-four InDels that were in the loop regions of proteins. We generated those InDels computationally and performed rigidity analysis on the resulting structures. We showed that we can generate low-energy InDels that are structurally very similar to the experimental PDB InDel mutant structures.

Our rigidity analysis proved that we often produce computationally generated InDels with rigidity properties – i.e. - location and size of rigid clusters vs. flexible regions – that are similar to the rigidity properties of the corresponding experimental InDel mutant structures in the PDB. Statistical analysis of the rigidity of the computational *in silico* and experimental (PDB) mutants showed that both reveal statistically significant differences from the wildtype, while having similar rigidity properties to one another. Here we would like to address that the statistical analysis is currently performed on the set of proteins that were used for the purpose of this preliminary study. However, once more data is discovered, the analysis will be repeated in order to reevaluate the findings.

This initial study shows that our method has the ability to generate InDels and bridge the gap between the importance of InDels and their lack of experimental and computational availability.

Acknowledgements

This research was funded in part by an NSF grants IIS-2031260 and IIS-2031283 of Haspel and Jagodzinski. We are grateful to computer science students Anna Zivkovic and Sean McCulloch from Western Washington University for their contributions to the first version of the software integrating the *in silico* InDel generation computation step with the rigidity analysis portions of the framework.

References

[1] Maoxuan Lin, Sarah Whitmire, Jing Chen, Alvin Farrel, Xinghua Shi, and Jun-tao Guo. Effects of short indels on protein structure and function in human genomes. *Scientific Reports*, 7(1):9313, Aug 2017.

[2] Pagel KA, Antaki D, Lian A, Mort M, Cooper DN, and et al. Sebat J. Pathogenicity and functional impact of non-frameshifting insertion/deletion variation in the human genome. *Plos Comp. Biol.*, 15(6):e1007112, 2019.

[3] Jennifer K. Sehn. Chapter 9 - insertions and deletions (indels). In Shashikant Kulkarni and John Pfeifer, editors, *Clinical Genomics*, pages 129–150. Academic Press, Boston, 2015.

[4] RyangGuk Kim and Jun-tao Guo. Systematic analysis of short internal indels and their impact on protein folding. *BMC Structural Biology*, 10(1):24, Aug 2010.

[5] Sara Light, Rauan Sagit, Oxana Sachenkova, Diana Ekman, and Arne Elofsson. Protein Expansion Is Primarily

due to Indels in Intrinsically Disordered Regions. *Molecular Biology and Evolution*, 30(12):2645–2653, 09 2013.

[6] Zheng Zhang, Jie Huang, Zengfang Wang, Lushan Wang, and Peiji Gao. Impact of Indels on the Flanking Regions in Structural Domains. *Molecular Biology and Evolution*, 28(1):291–301, 07 2010.

[7] Stephane Emond, Maya Petek, Emily J. Kay, Brennen Heames, Sean R. A. Devenish, Nobuhiko Tokuriki, and Florian Hollfelder. Accessing unexplored regions of sequence space in directed enzyme evolution via insertion/deletion mutagenesis. *Nature Communications*, 11(1):3469, Jul 2020.

[8] Romain A Studer, Benoit H Dessailly, and Christine A Orengo. Residue mutations and their impact on protein structure and function: detecting beneficial and pathogenic changes. *Biochemical journal*, 449(3):581–594, 2013.

[9] Pravech Ajawatanawong and Sandra L Baldauf. Evolution of protein indels in plants, animals and fungi. *BMC evolutionary biology*, 13(1):1–15, 2013.

[10] Zheng Zhang, Yuxiao Wang, Lushan Wang, and Peiji Gao. The combined effects of amino acid substitutions and indels on the evolution of structure within protein families. *PloS one*, 5(12):e14316, 2010.

[11] HA Lewis, C Wang, X Zhao, Y Hamuro, K Conners, MC Kearins, F Lu, JM Sauder, KS Molnar, SJ Coales, et al. Structure and dynamics of nbd1 from cftr characterized using crystallography and hydrogen/deuterium exchange mass spectrometry. *Journal of molecular biology*, 396(2):406–430, 2010.

[12] Elisa Donnard, Paula F Asprino, Bruna R Correa, Fabiana Bettoni, Fernanda C Koyama, Fabio CP Navarro, Rodrigo O Perez, John Mariadason, Oliver M Sieber, Robert L Strausberg, et al. Mutational analysis of genes coding for cell surface proteins in colorectal cancer cell lines reveal novel altered pathways, druggable mutations and mutated epitopes for targeted therapy. *Oncotarget*, 5(19):9199, 2014.

[13] Prathima Iengar. An analysis of substitution, deletion and insertion mutations in cancer genes. *Nucleic acids research*, 40(14):6401–6413, 2012.

[14] Zhe Liu, Huanying Zheng, Huifang Lin, Mingyue Li, Runyu Yuan, Jinju Peng, Qianling Xiong, Jiufeng Sun, Baisheng Li, Jie Wu, et al. Identification of common deletions in the spike protein of severe acute respiratory syndrome coronavirus 2. *Journal of virology*, 94(17):e00790–20, 2020.

[15] Michael Hsing and Artem Cherkasov. Indel pdb: a database of structural insertions and deletions derived from sequence alignments of closely related proteins. *BMC bioinformatics*, 9(1):1–12, 2008.

[16] Pravech Ajawatanawong, Gemma C Atkinson, Nathan S Watson-Haigh, Bryony MacKenzie, and Sandra L Baldauf. Seqfire: a web application for automated extraction of indel regions and conserved blocks from protein multiple sequence alignments. *Nucleic acids research*, 40(W1):W340–W347, 2012.

[17] Enrique E Abola, Frances C Bernstein, and TF Koetzle. The protein data bank. in neutrons in biology, 1984.

[18] UniProt Consortium. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1):D204–D212, 2015.

[19] Shota Shiga, Masaru Yamanaka, Wataru Fujiwara, Shun Hirota, Shuichiro Goda, and Koki Makabe. Domain-swapping design by polyproline rod insertion. *ChemBioChem*, 20(19):2454–2457, 2019.

[20] Emmanuel Boutet, Damien Lieberherr, Michael Tognolli, Michel Schneider, and Amos Bairoch. Uniprotkb/swiss-prot. In *Plant bioinformatics*, pages 89–112. Springer, 2007.

[21] Andrew Leaver-Fay, Michael Tyka, Steven M Lewis, Oliver F Lange, James Thompson, Ron Jacak, Kristian W Kaufman, P Douglas Renfrew, Colin A Smith, Will Sheffler, et al. Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. *Methods in enzymology*, 487:545–574, 2011.

[22] Ibrahim Al-Bluwi, Marc Vaisset, Thierry Siméon, and Juan Cortés. Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods. *BMC structural biology*, 13(1):1–14, 2013.

[23] Rachel Kolodny, Leonidas Guibas, Michael Levitt, and Patrice Koehl. Inverse kinematics in biology: the protein loop closure problem. *The International Journal of Robotics Research*, 24(2-3):151–163, 2005.

[24] Juan Cortés and Thierry Siméon. Sampling-based motion planning under kinematic loop-closure constraints. In *Algorithmic Foundations of Robotics VI*, pages 75–90. Springer, 2004.

[25] RosettaCommons Org. KIC Tutorial generalized kinematic closure 1, 2017.

[26] Peggy Yao, Ankur Dhanik, Nathan Marz, Ryan Propper, Charles Kou, Guanfeng Liu, Henry Van Den Bedem, Jean-Claude Latombe, Inbal Halperin-Landsberg, and Russ B Altman. Efficient algorithms to explore conformation spaces of flexible protein loops. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(4):534–545, 2008.

[27] Po-Ssu Huang, Yih-En Andrew Ban, Florian Richter, Ingemar Andre, Robert Vernon, William R Schief, and David Baker. Rosettaremodel: a generalized framework for flexible backbone protein design. *PloS one*, 6(8):e24109, 2011.

[28] Daniel J Mandell, Evangelos A Coutsias, and Tanja Kortemme. Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nature methods*, 6(8):551–552, 2009.

[29] Amelie Stein and Tanja Kortemme. Improvements to robotics-inspired conformational sampling in rosetta. *PloS one*, 8(5):e63090, 2013.

[30] Junichi Higo and Masayoshi Nakasako. Hydration structure of human lysozyme investigated by molecular dynamics simulation and cryogenic x-ray crystal struc-

ture analyses: on the correlation between crystal water sites, solvent density, and solvent dipole. *Journal of computational chemistry*, 23(14):1323–1336, 2002.

[31] Andrey Karshikoff, Lennart Nilsson, and Rudolf Ladenstein. Rigidity versus flexibility: the dilemma of understanding protein thermal stability. *The FEBS journal*, 282(20):3899–3917, 2015.

[32] Filip Jagodzinski, Jeanne Hardy, and Ileana Streinu. Using rigidity analysis to probe mutation-induced structural changes in proteins. *Journal of bioinformatics and computational biology*, 10(03):1242010, 2012.

[33] Ramin Dehghanpoor, Evan Ricks, Katie Hursh, Sarah Gunderson, Roshanak Farhoodi, Nurit Haspel, Brian Hutchinson, and Filip Jagodzinski. Predicting the effect of single and multiple mutations on protein structural stability. *Molecules*, 23(2):251, 2018.

[34] N. Fox, F. Jagodzinski, Y. Li, and I. Streinu. KINARI-web: A server for protein rigidity analysis. *Nucleic Acids Research*, 39 (Web Server Issue):W177–W183, 2011.

[35] E. Andersson, R. Hsieh, H. Szeto, R. Farhoodi, F. Jagodzinski, and N. Haspel. Assessing how multiple mutations affect protein stability using rigid cluster size distributions. In *proc. of IEEE-ICCABS (International Conference on Computational Advances in Bio and Medical Sciences)*, October 2016.

[36] Clark Wissler. The spearman correlation formula. *Science*, 22(558):309–311, 1905.