On the Stochastic Roadmap to Model Functionally-related Structural Transitions in Wildtype and Variant Proteins

Kevin Molloy*, Rudy Clausen*, Amarda Shehu *†‡
*Dept. of Computer Science, †Dept. of Bioengineering, ‡School of System Biology
George Mason University, Fairfax, Virginia
amarda@gmu.edu

***Abstract*—Evidence is emerging that the role of protein structure in disease needs to be rethought. While many proteinopathies are caused by sequence mutations removing the ability of a protein to assume a specific structure, some of the most complex human diseases are not so easily explained. Mutations may not invalidate structures populated by the wildtype protein but instead affect the rate at which the protein switches between structures. Modeling structural transitions and estimating transition rates in wildtype and variants is central to a better understanding of the molecular basis of disease. Building on seminal work on the stochastic roadmap simulation framework, this paper investigates an efficient algorithmic realization of this framework to model structural transitions in wildtype and variants of an oncogene. Our results indicate that the algorithm is able to extract useful kinetic information and elucidates the role of structure in how sequence mutations affect protein function.**

I. Introduction

The increasingly accepted view of proteins as inherently dynamic systems [16] is raising questions on the role of protein structure in diseases that are proteinopathies. The simplified view of proteins assuming a unique structure to carry out their biological activity [1] allows explaining some protein conformational diseases [32]. In these, the protein is unable to assume its designated function-carrying structure due to internal perturbations (sequence mutations) or external ones in the environment (cellular stress). However, increasing evidence is emerging on enzymes and other proteins making use of a menu of stable or semi-stable structures to modulate their function and participate in numerous complex chemical processes in the cell [5]. Both experiment and computation have shown that many proteins switch between structures, undergoing productive structural displacements of less than an angstrom or on the order of a few angstroms [6].

In light of this dynamic view of proteins, it is unclear how mutations that cause or participate in disease affect structure, which is the intermediate link in the relationship between protein sequence and function. There is evidence that some of the most complex human diseases, including cancer, do not arise from the inability of a protein variant to occupy a specific structure but rather from changes to the rate at which the variant transitions between thermodynamically-stable or semi-stable structures [11]. Due to such evidence and the inability of wet-laboratory techniques to elucidate structural transitions, it is central to explore computational techniques to model such transitions and extract kinetic information, such as transition rates. While typically kinetic data can be obtained with methods based on Molecular Dynamics (MD), such methods tend to be prohibitively computationally expensive [13]. Many MD simulations need to be launched to sample an ensemble of trajectories over which to calculate statistics of interest.

Seminal work by Latombe and colleagues proposed the employment of the probabilistic roadmap, a construct that emerged in algorithmic robotics to plan robot motions and was adapted to model molecular motions [29]. Soon after, the stochastic roadmap simulation (SRS) framework [3] was formulated to replace sampling numerous Monte Carlo (MC) or MD trajectories with a graph or roadmap whose edges capture transitions between sampled structures. Organizing sampled structures into structural states and thus treating the roadmap as a Markov model allowed calculating kinetic data without launching a single MC or MD simulation [8].

Since its introduction, realizations of SRS have been pursued primarily to model folding and unfolding events of small proteins or RNA [31, 34, 35, 33]. Tree-based variations to model other structural transitions have also been proposed, a review of which can be found in [28]. More powerful Markov-based approaches integrating short MD trajectories have been investigated to improve accuracy of the modeled kinetics in folding [30]. Realizations of SRS to approximate the kinetics of general and potentially large structural transitions in dynamic proteins have been scarce [21] due to several challenges, including efficient yet dense sampling in the space of interest.

In this paper we investigate an efficient algorithmic realization of the SRS framework to model and then compare structural transitions in wildtype and variants of the Ras oncogene. While the algorithm faithfully follows the SRS framework for extracting statistics, its construction of the roadmap is lazy, and its sampling of structures is performed a priori through a powerful evolutionary algorithm (EA) sampling energy basins in the energy landscape of a protein. Details on the EA are provided elsewhere [10]. Here we focus on the construction of the stochastic roadmap and the calculation of interesting statistics, such as average number of transitions to structural states of interest. Comparison of these statistics between wildtype and variants of Ras demonstrate the ability

of the proposed SRS-based algorithm to elucidate changes to structural transition rates upon sequence mutations and so explain how mutation affects function in dynamic proteins.

II. Methods

The proposed algorithm follows the SRS framework. Briefly, it proceeds in three stages. The first stage samples structures in the search space of interest. The second organizes these structures into structural states. The third embeds a roadmap over the states. We now proceed to relate details.

A. Stage I: Sampling

One of the key challenges with adaptations of roadmap-based methods for molecular structure and motion computation lies in the sampling stage. Sampling needs to be dense and focus on the relevant regions of the structure space. In this paper, we employ an evolutionary algorithm (EA) to obtain a dense ensemble of structures representing local energy minima in the structure space of interest. Though a detailed description of this EA is beyond the scope of this paper, we provide here a brief summary, focusing on its salient algorithmic ingredients.

EAs are investigated in detail in our lab in diverse protein modeling scenarios, including *de novo* structure prediction [22, 26] and protein-protein docking [15, 25]. The EA we employ here has been recently proposed [10] to further populate the structure space of a protein for which many experimental structures already exist in the Protein Data Bank (PDB) [4]. Briefly, the EA leverages the abundance of experimentally-available structures to define the structure space of interest in a lower-dimensional embedding. The latter is obtained through Principal Component Analysis of CA-traces (using only CA atoms to represent structures) of available structures for a protein. While PCA is generally not guaranteed to be effective, the EA only proceeds if at least 50% of the variance can be captured with the top two principal components (PCs). This is the case with the protein system we have chosen to investigate in this paper. The EA directly searches in the low-dimensional PC map of m dimensions, ensuring that m PCs are sufficient to capture 90% of the variance in the original structure data.

Starting with an initial population of p structures built on the experimentally-available ones, reproductive operators are used to generate child structures (in a CA trace representation) from parents in the PC map, using sampled perturbations along the available PCs. A multiscaling procedure maps a child structure to an all-atom structure representative of a local energy minimum. The procedure first reconstructs a backbone from the CA trace of a child, adds side chains, and minimizes the entire resulting all-atom structure using the Rosetta *relax* protocol (keeping backbone heavy atoms fixed). This procedure ensures that structures obtained by the EA are minima of the all-atom Rosetta *score12* energy function [18]. The resulting minima structures compete with neighboring parents based on their energies, and p winners become parents of the next generation. This proceeds for a certain number g of generations. It is worth noting that searching in a PC-based embedding and making use of multiscaling have been previously analyzed in detail in the context of a robotics-inspired (tree-based) search algorithm [9], and these components are integrated in the recently proposed EA [10] we employ in the sampling stage here. The ensemble Ω of structures fed to stage II of the SRS-based algorithm in this paper consists of all the populations of local minima obtained by the EA across all its g generations.

B. Stage II: Organizing Structures into Structural States

The ensemble Ω potentially contains many structures that are geometrically similar to one another. Therefore, in this stage, the structures in Ω are grouped into structural states both to remove redundancy and to allow constructing a roadmap over these states that can then be treated and analyzed as a Markov state model. We employ a simple unsupervised clustering algorithm, leader clustering [14], to efficiently group structures into states. That is, a structural state is a cluster.

The leader clustering algorithm has the benefit of not having to specify the number of clusters/states a priori. Its results are dependent on the order in which the data is processed. In this paper, we use a sorted order, ordering first all the structures in the Ω ensemble by their Rosetta energies. This ordering allows the first structure mapped to a new cluster to be the lowest-energy structure over all others that will be mapped to that same cluster. The algorithm proceeds in the sorted order, mapping a structure to one of the existing clusters if its distance to the cluster representative is below a specified cluster radius. Otherwise, a new cluster is created with the unmapped structure as its representative. The algorithm proceeds until all structures have been processed, resulting in a list of $C_1, \ldots, C_l$ clusters/states. The decision on what distance function to use is important. Here we employ least Root Mean Squared Deviation (lRMSD), which is a popular dissimilarity measure to compare protein structures [20]. We do so over only CA atoms of a structure; that is, we use CA lRMSD. We experiment with different values of cluster radii, as presented in the Results section.

C. Stage III: Roadmap Construction

Roadmap construction proceeds over the identified clusters. The roadmap is encoded as a weighted directed graph $G = (V, E)$. A vertex $v \in V$ is created for each of the clusters identified in stage II; that is, vertices encode states over the sampled structure space. Edges are added to the roadmap as follows. Each vertex is connected to up to k_{nn} of its nearest neighbors that are within an ϵ_{nn} CA lRMSD of v. Since vertices correspond to structural states/clusters, the lRMSD comparison is conducted between the cluster representatives. When a vertex u is deemed to be a neighbor of v that passes the k_{nn} and ϵ_{nn} criteria, two edges are added to the roadmap, (u, v) and (v, u). To improve the connectivity of the roadmap, a final pass across all connected components is performed, adding an edge when the two components can be merged (subject to the same ϵ_{nn} CA lRMSD constraint). Edges are weighted based on the energetic difference between the states the vertices they connect encode. For a directed edge (u, v), its weight P_{uv} measures the probability of a direct transition from

u to v. We assign edge weights following closely the original formulation of the SRS in [3], per the following equations:

$$P_{uv} = \begin{cases} (1/|N_u|) \cdot e^{\frac{-\Delta E_{uv}}{\alpha}} & \text{if } \Delta E_{uv} > 0 \\ 1/|N_u| & \text{otherwise} \end{cases}$$

$$P_{uu} = 1 - \sum_{u \neq v} P_{uv}$$

For each vertex v, $|N_v|$ represents the number of outgoing edges from v excluding the edge back to itself. The $e^{\frac{-\Delta E_{uv}}{\alpha}}$ factor mimics the Metropolis criterion for accepting the energetic transition from state u to state v. Note that $\Delta E_{uv} = E(v) - E(u)$. There are two important decisions that need to be made. First, since vertices here encode structural states, how is the energy of a state measured? Second, how is a reasonable value for the α parameter estimated? Our specific choices for these two design decisions can be considered adaptations of the original SRS formulation on weighting direct transitions.

a) Energy of a State: Theoretically, if the states correspond to energetic states, one should measure $E(u)$ as the free-energy F of the state u. This can be estimated, in theory, as $F(u) = \langle E \rangle_u - \alpha \cdot ln(|C_u|)$, where $\langle E \rangle_u$ captures the average energy over all structures in the state u, and $|C_u|$ measures the number of structures in u. However, in practice, an accurate estimate requires a theoretically-sound definition of a state and is the subject of our future research. In this paper, we pursue two directions. One is to measure the energy of a state as the average over energies of structures in the cluster corresponding to that state. The other is to use the energy of the cluster representative, which is the lowest-energy structure in a cluster due to the energy-sorted order in which structures are processed in the clustering algorithm described above.

b) Effective Temperature: The scaling parameter α is our adaptation of the original equations appearing in [3]. We introduce this parameter instead of the $K_B T$ factor in [3] (where K_B is the Boltzmann constant and T refers to physical temperature) in order to capture an effective rather than a physical temperature. This is necessary, as the energy function employed in this paper is not physics-based but combines physics-based terms with knowledge-based ones. Determining the value for α is an important decision, and we employ here a simple analysis based on statistical mechanics. We measure the energetic variance over structures that the Rosetta score12 energy function reports to be in the same energy basin. We restrict our analysis over the distribution of structures obtained by the Rosetta relax protocol when minimizing the same crystal structure many times. The protocol is based on simulated annealing, so different structures can be obtained, thus providing a view of the basin where the Rosetta energy function maps a given crystal structure. We conduct this analysis various times, over different crystal structures (corresponding to the on and off states described in the Experimental Setup in the Results section) and observe a variance of $6-7$ energy units on average. Based on a statistical mechanics treatment, structures in the same basin should exchange into one another with high probability. Let us refer to the latter as a target probability tprob. Therefore, solving the equation $e^{-6/\alpha} = \mathrm{tprob}$ for α provides us with a reasonable estimate for the effective temperature. We note that the actual value for α is dependent on the energy function employed and requires that a target probability be specified, but the process is general.

Each edge in the stochastic roadmap G now encodes a potential transition between two structural states. In this work, we employ a "lazy" strategy that avoids the computation of these transitions and instead focuses on the global connectivity. This has some similarities to the Lazy PRM[7]. We note, however, that foregoing a local planner is made possible here because of the stringent criterion of structural proximity ϵ_{nn} when considering connecting two vertices via an edge. This in itself exploits the dense structural sampling afforded by the EA employed in stage I.

We note that by construction G consists of a set of strongly connected components (SCCs); when $\epsilon_{nn} = \infty$, G consists of a single SCC. As demonstrated in [3], a random walk in G can be interpreted as a discretized version of a Monte Carlo trajectory. More importantly, various analyses can be conducted over the roadmap to obtain path-ensemble averages without launching Monte Carlo simulations, as the roadmap encodes multiple such trajectories.

D. Roadmap Analysis

Treating the constructed stochastic roadmap as a graph allows using path search algorithms to obtain paths connecting structural states of interest. Treating the roadmap as a Markov state model allows using transition state theory to obtain measurements approximating kinetic quantities of interest.

1) Querying the Roadmap: As demonstrated in the original proposal of the PRM method in [19], the roadmap can be queried given two states of interest. Dijkstra's algorithm can be used to obtain a shortest path. Here, edges are weighted by probabilities of transition, but negatives of logarithms of these probabilities can be employed to obtain a minimum-cost path. In addition to such a path, more information can be obtained by analyzing not just one path but several. Yen's K-shortest paths algorithm [27] can be employed for this purpose.

2) Treating the Roadmap as a Markov State Model: The roadmap G can be treated as a Markov state model encoding the stochastic behavior of the system being studied. In this paper, we use the roadmap to model the structural transitions between functionally-relevant states of a protein and understand how these transitions are affected by sequence mutations. For this purpose, the roadmap G is analyzed to determine the expected number of transitions employed by a protein system to switch from one structural state to another.

Recall that structural states are vertices in the vertex set V in our roadmap G. For each vertex $v_i \in V$, one can utilize first-step analysis theory to measure the expected number of transitions t_i from vertex v_i to some specific vertex of interest. As demonstrated in[2], random walks need not be performed to obtain such a measure, as a closed-form solution can be computed via a linear solver. The formulation of t_i is recursive. Let us generalize and state that the goal is to measure the expected number of transitions from some vertex v_i to a set

of vertices $v_j \in A$, where A is a subset of V that does not include v_i (A is in an SCC). Then, provided that v_i and A are in the same SCC:

$$t_i = 1 + \sum_{v_j \in A} P_{ij} \cdot 0 + \sum_{v_j \notin A} P_{ij} \cdot t_j \quad \forall\ v_i \notin A$$

This results in a system of equations that is the same order as the number of vertices in the graph. Since clustering of structures into structural states reduces the number of vertices in the roadmap, an exact solver (as opposed to a slow-converging iterative solver) can be afforded, and that is what we employ in this paper to solve the linear system above algebraically and obtain t_i for all the vertices simultaneously.

In this paper, we are specifically interested in measuring the expected number of transitions from an "on" to an "off" state and vice versa, with these two states denoting specific structural states critical to the ability of Ras to function normally. By repeating the sampling, clustering, roadmap construction, and its analysis on different sequence variants of RAS, we then are able to compare the expected number of transitions between these two states of interest in the wildtype versus disease-participating variants of RAS.

a) Implementation Details: The algorithm is implemented C++. The EA in the sampling stage runs for g=100 generations, with p=500 structures in a population. Thus, the ensemble of structures Ω fed to the clustering stage contains $50,000$ structures. It takes 48 days of CPU time on a single 2.66 GHz Opteron processor with 24 GB of memory to obtain this ensemble. Various cluster radii are investigated in the clustering stage. For the results shown in this paper, a radius of 0.35Å is used, as it is observed that the number of clusters goes down from $46,193$ to $37,250$, to $26,461$, and to $17,547$ when the radius accordingly varies in $\{0.25, 0.3, 0.35, 0.4\}$Å. The clustering stage takes approximately 7 hours of CPU time. Parallelizing reduces the run time to just under 45 minutes on a 64 core AMD Opteron processor with 542 GB of memory. This same hardware is used to perform the roadmap construction and analysis, which each execute in approximately 30 minutes. While various values for k_{nn} and e_{nn} are investigated for how they affect connectivity, the results related here are obtained with k_{nn}=20 and ϵ_{nn}=0.66Å. In determining a reasonable value for the effective temperature α per the process described above, we err here on the conservative side and set tprob to 0.25 (we relate details with two different values, a conservative one and a more permissive one).

III. Results

A. Experimental Setup

Here we present results on the application of the proposed algorithm on the wildtype and Q61L variant of the Ras oncogene. Ras is a well-studied protein that regulates cell proliferation and whose variants which deregulate activity are involved in over 25% of human cancers [17]. The native activity of Ras is to switch between an ON/reactant (GTP-bound) and an OFF/product (GDP-bound) state. These two states have been characterized in the wet laboratory and can be found under structures with PDB ids 1qra and 4q21, respectively. We show these structures side by side in Figure 1. The CA lRMSD between these structures is 1.5Å, but changes are largely localized on two loop regions, switch I and switch II (which our previous analysis of PCA for Ras is able to capture [9]). How variations in the Ras sequence affect its capacity for switching between states is the focus of much research and is the reason we apply our SRS-based algorithm here.

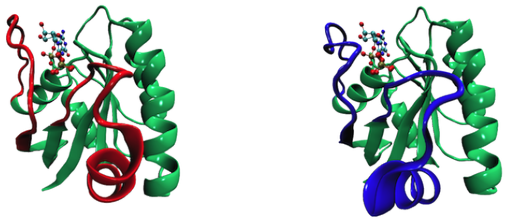

Fig. 1. Left: A representative of the ON (GTP-bound) state of Ras (PDB id: 1qra). Right: A representative of the OFF (GDP-bound) state (PDB id: 4q21). The reactant (GTP) and product (GDP) are shown, as well. The two loop regions that undergo a structural change in the ON to OFF transition and (reverse) are shown color-coded in red (left) and blue (right).

The reduced space over which the sampling stage operates is obtained via PCA on 46 (wildtype and variant) structures extracted for Ras from the PDB (details on the data collection step can be found in [9]). The SRS-based algorithm is run twice, once on the wildtype sequence and once on the disease-participating variant (Q61L). It is important to note that, while the PCs are the same in each setting, the EA algorithm obtains different structural ensembles, as the initial structures are threaded onto the sequence of study, and thus mapped by the multiscaling procedure to minima of different sequence-dependent energy surfaces. Thus, the results of the SRS-based algorithm are dependent on the sequence used and can be used to draw comparisons between the wildtype and variants to determine how sequence mutations affect transitioning between the ON and OFF states.

The structure in the PDB entry 1qra is considered a representative of the ON state of Ras, whereas 4q21 is a representative of the OFF state. These PDB-obtained structures are each minimized 500 times with the Rosetta relax protocol (the protocol is stochastic), and the resulting structures are added to the Ω ensemble. After the clustering, the cluster containing the most minimized structures of 1qra is labeled the ON state, whereas the cluster containing the most minimized structures of 4q21 is labeled the OFF state.

B. Roadmap Analysis on Wildtype and Q61L Variant

We apply the analysis techniques discussed in section II-D to the roadmaps created for the wildtype and Q61L sequences. The minimum cost paths between the ON and OFF states for each sequence are computed and analyzed first. Column 3 in Table I shows the total cost of each of these paths. Comparison of these values shows that the ON $\rightarrow$ OFF structural transition is more costly than the OFF $\rightarrow$ ON one for both the wildtype and Q61L. However, both transitions have higher cost in the Q61L variant, indicating a significant change of the energy landscape upon this mutation.

The minimum-cost paths for each of these transitions in the wildtype are shown in Figure 2. For ease of visualization,

TABLE I

THE MINIMUM-COST PATHS AND THE EXPECTED NUMBER OF TRANSITIONS ARE SHOWN FOR THE STRUCTURAL TRANSITIONS BETWEEN THE ON AND OFF STATES IN BOTH THE WILDTYPE AND Q61L VARIANTS.

Sequence	Transition	Min Cost	Exp. Nr. Trans
WT	OFF $\rightarrow$ ON	12.9	3.4×10^{8}
	ON $\rightarrow$ OFF	16.5	3.9×10^{10}
Q61L	OFF $\rightarrow$ ON	20.9	1.9×10^{12}
	ON $\rightarrow$ OFF	24.3	3.8×10^{14}

the paths are mapped onto the top two PCs. The color scheme follows the energy variance. Figure 2 shows that both structural transitions go over an energy barrier, as also reflected in the costs shown for the wildtype sequence in Table I. Moreover, the ON $\rightarrow$ OFF transition spends more time getting out of a deeper and wider ON basin onto the OFF basin.

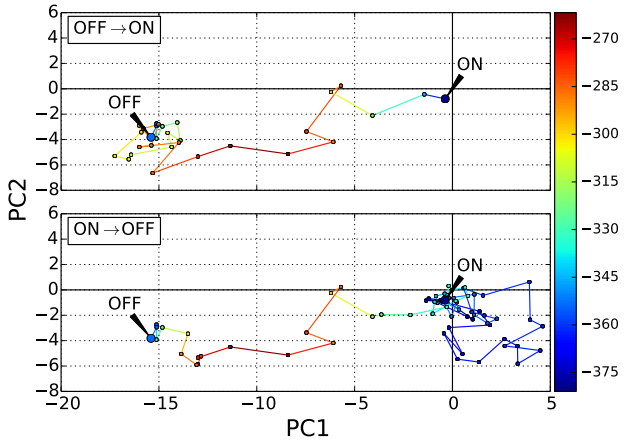

Fig. 2. The minimum cost paths (in terms of energy) are shown for the wildtype sequence between the OFF and ON states. This plot is rendered in the PCA space created by our EA algorithm for sampling.

The detailed energetic profiles of the minimum-cost paths for the ON $\rightarrow$ OFF transition in the wildtype and Q61L variant are shown in Figure 3. The Rosetta all-atom energy is shown for each vertex in these paths, but the path lengths are normalized to allow a direct comparison between the two sequences. Figure 3 clearly shows that the Q61L mutation magnifies the energy barrier that Ras has to cross in the ON $\rightarrow$ OFF structural transition. These results are in qualitative agreement with other studies [12] and allow concluding that the transition from the ON to the OFF state is made substantially more difficult upon the Q61L mutation in Ras. It is important to note that the mutation does not affect the stability of the ON and OFF structural states, since the potential energies of the corresponding states remain the same between the wildtype and variant.

Finally, the first-step analysis is applied to measure and compare the expected number of transitions in each setting. These results are related in column 4 in Table I. Comparison of these results for the wildtype sequence shows that the expected number of transitions to allow switching from the ON to the OFF state is two orders of magnitude higher than from the OFF to the ON state. This also holds for the Q61L variant, though switching from ON to OFF and vice versa becomes more difficult in the variant than in the wildtype.

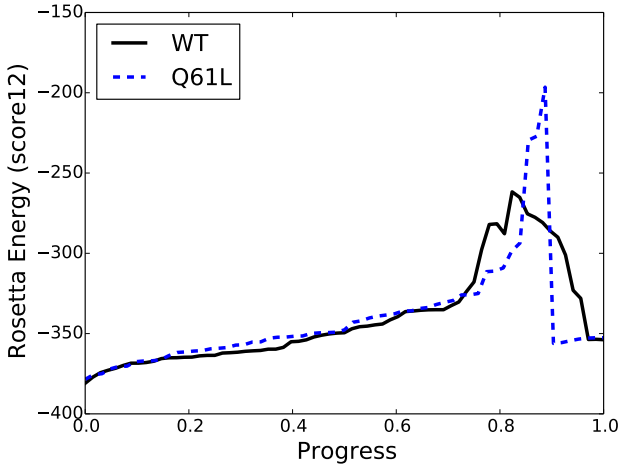

Fig. 3. The energetic profile of the minimum cost paths when transitions from the ON state to the OFF state for the wild type and Q61L mutant sequences.

Taken altogether, these results suggest that a careful realization of the SRS framework may allow a more detailed understanding of the role of sequence mutations in misfunction. In our particular application to the wildtype and Q61L variant of Ras, the results support the hypothesis that the Q61L mutation does not remove the ON and OFF basins from the energy landscape but instead slows down the switching of Ras between these states.

IV. SUMMARY

This paper has proposed an efficient algorithmic realization of the SRS framework to model structural transitions in dynamic proteins that are known to be conformational switchers and are involved in proteinopathies. Application on sequence variants of Ras shows promising results. Future work will continue to investigate the algorithmic richness of the SRS framework in order to improve both accuracy and efficiency in protein structure modeling for the purpose of unraveling the role of protein structure in proteinopathies. Possible directions include incorporating estimates of free energies of structural states in the calculation of transition probabilities, as well as employing different energy functions in a comparative setting to estimate the generalizability of the results.

ACKNOWLEDGMENTS

Many of these experiments were run on ARGO, a research computing cluster provided by the Office of Research Computing at George Mason University, VA (URL: http://orc.gmu.edu). Funding for this work is provided in part by the National Science Foundation (Grant No. 1016995 and CAREER Award No. 1144106) and the Thomas F. and Kate Miller Jeffress Memorial Trust Award.

REFERENCES

[1] C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[2] M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu, and J.-C. Latombe. Stochastic roadmap simulation: an efficient representation and algorithm for analyzing

molecular motion. *J. Comp. Biol.*, 10(3-4):257–281, 2003.

[3] M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu, and J.-C. Latombe. Stochastic roadmap simulation: an efficient representation and algorithm for analyzing molecular motion. *J. Comp. Biol.*, 10(3-4):257–281, 2003.

[4] H. M. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide Protein Data Bank. *Nat. Struct. Biol.*, 10(12):980–980, 2003.

[5] D. D. Boehr, D. McElheny, J. Dyson, and P. E. Wright. The dynamic energy landscape of dihydrofolate reductase catalysis. *Science*, 313(5793):1638–1642, 2006.

[6] D. D. Boehr, R. Nussinov, and P. E. Wright. The role of dynamic conformational ensembles in biomolecular recognition. *Nature Chem Biol*, 5(11):789–96, 2009.

[7] R. Bohlin and Lydia E. Kavraki. Path planning using lazy prm. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 521–528, San Fransisco, CA, April 2000. IEEE Press, IEEE Press. doi: 10.1109/ROBOT.2000.844107.

[8] T. H. Chiang, D. Hsu, and Latombe. J. C. Markov dynamic models for long-timescale protein motion. *Bioinformatics*, 26(12):269–277, 2010.

[9] R. Clausen and A. Shehu. Exploring the structure space of wildtype ras guided by experimental data. In *ACM Conf on Bioinf and Comp Biol Workshops (BCBW)*, pages 757–764, Washington, D. C., September 2013.

[10] R. Clausen and A. Shehu. A multiscale hybrid evolutionary algorithm to obtain sample-based representations of multi-basin protein energy landscapes. In *ACM Conf on Bioinf and Comp Biol (BCB)*, Newport Beach, CA, September 2014. under review.

[11] A. Fernández-Medarde and E. Santos. Ras in cancer and developmental diseases. *Genes Cancer*, 2(3):344–358, 2011.

[12] Alemayehu A. Gorfe, Barry J. Grant, and J. Andrew McCammon. Mapping the nucleotide and isoform-dependent structural and dynamical features of Ras proteins. *Structure*, 16(6):885–896, 2008.

[13] B. J. Grant, A. A. Gorfe, and J. A. McCammon. Ras conformational switching: Simulating nucleotide-dependent conformational transitions with accelerated molecular dynamics. *PLoS Comp Biol*, 5(3):e1000325, 2009.

[14] J.A. Hartigan. *Clustering Algorithms*. John Wiley and Sons, New York, 1975.

[15] I. Hashmi and A. Shehu. Informatics-driven protein-protein docking. In *ACM Conf on Bioinf and Comp Biol Workshops (BCBW)*, pages 772–779, Washington, D. C., September 2013.

[16] K. Jenzler-Wildman and D. Kern. Dynamic personalities of proteins. *Nature*, 450:964–972, 2007.

[17] Antoine E. Karnoub and Robert A. Weinberg. Ras oncogenes: split personalities. *Nature Reviews Molecular Cell Biology*, 9:517–531, 2008.

[18] Kristian W. Kaufmann, Gordon H. Lemmon, Samuel L. DeLuca, Jonathan H. Sheehan, and Jens Meiler. Practically useful: What the rosetta protein modeling suite can do for you. *Biochemistry*, 49(14):2987–2998, 2010.

[19] L. E. Kavraki, P. Svetska, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 12(4):566–580, 1996.

[20] A. D. McLachlan. A mathematical procedure for superimposing atomic coordinates of proteins. *Acta Crystallogr. A.*, 26(6):656–657, 1972.

[21] K. Molloy and A. Shehu. A probabilistic roadmap-based method to model conformational switching of a protein among many functionally-relevant structures. In *Intl Conf on Bioinf and Comp Biol (BICoB)*, Las Vegas, NV, 2014.

[22] B. Olson and A. Shehu. Efficient basin hopping in the protein energy surface. In *IEEE Intl Conf on Bioinf and Biomed*, Philadelphia, PA, October 2012. 119-124.

[25] B. Olson, I. Hashmi, K. Molloy, and A. Shehu. Basin hopping as a general and versatile optimization framework for the characterization of biological macromolecules. *Advances in AI J*, 2012(674832), 2012.

[26] B. Olson, K. A. De Jong, and A. Shehu. Off-lattice protein structure prediction with homologous crossover. In *Conf on Genetic and Evolutionary Computation (GECCO)*, New York, NY, 2013. ACM.

[27] M. Pascoal and E. Martins. A new implementation of Yen's ranking loopless algorith. *Quart J of the Belgian, French and Italian Oper Res Soc*, 1(2):121–133, 2003.

[28] A. Shehu. Probabilistic search and optimization for protein energy landscapes. In S. Aluru and A. Singh, editors, *Handbook of Computational Molecular Biology*. Chapman & Hall/CRC Computer & Information Science Series, 2013.

[29] A. P. Singh, J.-C. Latombe, and D. L. Brutlag. A motion planning approach to flexible ligand binding. In R. Schneider, P. Bork, D. L. Brutlag, J. I. Glasgow, H.-W. Mewes, and R. Zimmer, editors, *Proc Int Conf Intell Sys Mol Biol (ISMB)*, volume 7, pages 252–261, Heidelberg, Germany, 1999. AAAI.

[30] N. Singhal, C. D. Snow, and V. S. Pande. Using path sampling to build better markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *J. Chem. Phys.*, 121(1):415–425, 2004.

[31] G. Song and N. M. Amato. A motion planning approach to folding: From paper craft to protein folding. *IEEE Trans. Robot. Autom.*, 20(1):60–71, 2004.

[32] C. Soto. Protein misfolding and neurodegeneration. *JAMA Neurology*, 65(2):184–189, 2008.

[33] L. Tapia, S. Thomas, and N. Amato. A motion planning approach to studying molecular motions. *Communications in Information Systems*, 10(1):53–68, 2010.

[34] S. Thomas, G. Song, and N. M. Amato. Protein folding by motion planning. *J. Phys. Biol.*, 2(4):148, 2005.

[35] S. Thomas, X. Tang, L. Tapia, and N. M. Amato. Simulating protein motions with rigidity analysis. *J. Comput. Biol.*, 14(6):839–855, 2007.