A General, Adaptive, Roadmap-based Algorithm for Protein Motion Computation

Kevin Molloy* Amarda Shehu*†‡

*Department of Computer Science †Department of Bioengineering ‡School of Systems Biology

George Mason University, Fairfax, VA 22030, USA

Abstract

Precious information on protein function can be extracted from a detailed characterization of protein equilibrium dynamics. This remains elusive in wet and dry laboratories, as function-modulating transitions of a protein between functionally-relevant, thermodynamically-stable and meta-stable structural states often span disparate time scales. In this paper we propose a novel, robotics-inspired algorithm that circumvents time-scale challenges by drawing analogies between protein motion and robot motion. The algorithm adapts the popular roadmap-based framework in robot motion computation to handle the more complex protein conformation space and its underlying rugged energy surface. Given known structures representing stable and meta-stable states of a protein, the algorithm yields a time- and energy-prioritized list of transition paths between the structures, with each path represented as a series of conformations. The algorithm balances computational resources between a global search aimed at obtaining a global view of the network of protein conformations and their connectivity and a detailed local search focused on realizing such connections with physically-realistic models. Promising results are presented on a variety of proteins that demonstrate the general utility of the algorithm and its capability to improve the state of the art without employing system-specific insight.

Keywords

protein equilibrium dynamics; transition path sampling; conformation space; energy surface; roadmap-based algorithm; conformation sampling; stochastic optimization

I. Introduction

The intrinsic dynamic personality of proteins is by now not a surprise [1]. The jiggling and wiggling of atoms was mentioned early on by Feynman [2], and even predicted from first principles through polymer statistical thermodynamics by Cooper, Wolynes, and colleagues [3], [4]. Many studies have now revealed that protein dynamics under physiological conditions is harnessed into productive events in the cell [5]. In fact, for small and large proteins alike, functional modes of equilibrium dynamics have been detected [6], where concerted structural fluctuations mediate transitions between thermodynamically-stable and semi-stable structural states that allow participation in different molecular recognition events [7], resulting in function modulation [8].

A detailed characterization of protein equilibrium dynamics is important to advance our understanding of the relationship between protein dynamics and function, as well as aid drug discovery, protein-based sensor design, and

other protein engineering [9], [10]. While precious for understanding protein function, a detailed characterization of protein equilibrium dynamics remains challenging both in the wet and dry laboratories. Elucidating the detailed series of successive structures assumed by a protein in a transition involves disparate time scales. While atomic oscillations occur in the femto-pico second scale, concerted structural fluctuations and rearrangements may occur beyond the nano-second scale.

The disparate time scales pose a unique challenge in the wet laboratory. Thermodynamically-stable and semi-stable structural states at equilibrium correspond to broad and deep basins in the energy surface underlying the structure space of a protein [11]. A typical transition between such states is characterized by a long time where a protein diffuses in a basin before then making a sudden, quick jump to another basin; that is, the actual interesting event is too fast to be captured in the wet laboratory. While significant advances have been made by single-molecule, spectroscopy wet-laboratory techniques, in principle, such techniques cannot reveal the detailed series of structures in a transition event [12]. On the other hand, computational methods, while promising to aid in this regard, experience their own unique set of challenges.

The disparate time scales involved in protein transitions are a manifestation of the multitude of local minima in the protein energy surface. This challenge is known as the multiple minima issue, and methods seeking to compute transition paths that connect two given functionally-relevant structures often get stuck in local minima and do not reach the end structural state of the sought transition within a given computational budget. Methods that simulate molecular dynamics (MD) by integrating Newton's equation of motion on a finely-discretized time scale, essentially following the curvature of the energy surface, are particularly susceptible [13].

Other, non-MD methods follow a stochastic optimization process to navigate the energy surface via biased random walks. These methods typically implement Monte Carlo (MC) local search. While in principle promising a higher exploration capability than MD-based methods, MC-based methods also suffer from insufficient sampling due to the presence of local minima. As local search methods, they are typically enhanced with other algorithmic ingredients in order to escape local minima. A review of MD- and MC-based methods and their enhancements for navigating the rugged protein energy surfaces can be found in Ref. [14].

A third group of methods not based on the classic MD or MC frameworks is emerging as relevant for modeling protein dynamics. This group of methods, originally proposed in the robot motion planning community and known as robotics-inspired methods, expose similarities between the problem of computing protein transition paths and computing robot motions and rely on powerful analogies between protein and robot kinematics [15]. The essential idea is to obtain sample-based representations of valid regions in the space of variables used to represent a protein structure and embed samples in connectivity, graph-like structures that can be queried for paths connecting two desired structures. The variables of interest can be Cartesian coordinates, angles, or other collective variables elucidated from trivial or sophisticated, statistical analysis of known structures of a protein. From now on, we will refer to an instantiation over the defined variables as a *conformation*, and refer to the variable space navigated in search of transition paths as the conformation space. We do so in keeping with literature in computational structural biology and chemistry, where such an instantiation is typically referred to as a *conformer* or *conformation*. We note,

however, that the equivalent term in robotics and AI literature is *configuration*. In the context of protein transitions, a transition path is a series of successive conformations, starting with a conformation representing the start structure and ending with a conformation representing the goal structure.

Current robotics-inspired methods are tree-based or roadmap-based. Tree-based methods grow a tree search structure in conformation space from a given start to a given goal conformation. The growth of the tree is biased towards the goal conformation. As a result of the biasing, tree-based methods conduct efficient, albeit limited sampling of the conformation space. They are limited to finding one path from a given start to the goal conformation, a setting known as single-query. Tree-based methods need to be run multiple times to obtain various paths. However, the bias in the growth of the tree causes path correlations among runs.

Tree-based methods have seen broad applicability in modeling protein dynamics. They have been employed to sample transition paths in small peptides of less than a dozen amino acids and in large proteins of several hundred amino acids [16], [17], [18], [19], [20], [21]. However, in addition to the issue of high inter-run path correlations, existing tree-based methods cannot be readily employed for modeling transition paths in any given protein. Current methods rely heavily on specific insight and analysis on a protein at hand to determine a small, effective set of variables to define a conformation space, where biased sampling is likely to lead to transition paths. For instance, work in [18], [20] employs low-frequency modes elucidated from normal mode analysis (NMA) of structures, whereas work in [16], [17] focuses on angles that are different between the start and goal conformations. Work in [21] bundles consecutive backbone angles together, reducing the number of variables and employing the molecular fragment replacement technique to rapidly sample low-energy conformations. Work in [22] employs the more general backbone dihedral angles as variables, but circumvents the expected issue of dimensionality of the conformation space by focusing on peptides of a few amino acids.

Roadmap-based methods provide the ability to answer multiple queries through graph search algorithms on a constructed graph/roadmap of nearest-neighbor conformations. Unlike tree-based methods, where conformations are sampled as part of tree growth, roadmap-based methods detach the process of conformation sampling from that of embedding sampled conformations in a connectivity search structure, the roadmap. Roadmap-based methods for modeling protein dynamics adapt the popular Probabilistic RoadMap (PRM) framework proposed for robot motion computation in [23].

While in principle promising a broader view of the conformation space relevant for transitions, and as a result a more diverse set of paths from a single run, roadmap-based methods have traditionally been limited to modeling unfolding events in small (< 100 amino-acid long) proteins [24], [25], [26], [27], [28]. Several challenges limit their broad applicability. Significant computational resources can be spent sampling regions of the conformation space not relevant for the sought transition. On the other hand, sampling conformations in regions of interest is difficult with no a priori knowledge.

Another challenge concerns the realism of connections made between two nearest-neighbor conformations. Once two nearest neighbors are connected with an edge as part of the roadmap construction, the motion represented by that edge needs to be computed or realized through a local search technique known as a local planner. The local

planner needs to find intermediate conformations. Doing so may be exceptionally challenging, either because the planner may have to connect vertices of the roadmap far away in conformation space, if the sampling has not been dense, or vertices that are separated by a high-energy barrier in the energy surface. Significant computational time may be spent by local planners to realize all edges in the roadmap before being able to apply simple graph search algorithms to report paths connecting conformations of interest.

At present, there are no robotics-inspired methods that can be broadly and generally applied to model transitions spanning disparate time scales in small and large proteins alike. Yet, methods that do not have to rely on system-specific insight are useful to obtain a benchmark of their performance and set the stage for further algorithmic advances in robotics-inspired transition path sampling for proteins. In this paper, we address this issue by proposing such a general method.

We propose here the Stochastic Protein motIon Roadmap ALgorithm, referred to as *SPIRAL*. *SPIRAL* is an adaptation of the fuzzy PRM method originally introduced for robot manipulation planning [29]. *SPIRAL* is a sophisticated algorithm that operates within a limited computational budget and spends that budget in a priority-based scheme to realize promising paths. *SPIRAL* balances computational resources between a global search aimed at obtaining a global view of the network of protein conformations and their connectivity and a detailed local search aimed at realizing such connections.

In a departure from the common setup in transition path sampling, *SPIRAL* is designed to accommodate studies on transitions amongst possibly more than two structures. This setup allows additional applications on proteins known to assume a variety of stable and semi-structural states, some of which are studied here in this paper. *SPIRAL* exploits the knowledge of start and goal structures to focus sampling in their vicinity and avoid wasting resources sampling conformations of no relevance for the desired transition(s).

SPIRAL is designed to be general and not customized to a protein at hand in its selection of variables. The goal is to provide through *SPIRAL* a first-generation, general algorithm that can be used as a benchmark to further spur research on roadmap-based frameworks for sampling transition paths spanning disparate time and spatial scales.

II. Methods

SPIRAL consists of two main stages, sampling and roadmap building. The sampling stage generates a set of conformations/samples, which we refer to as the ensemble Ω. It is worth noting that Ω is not a thermodynamic ensemble but a set of conformations that satisfy specified constraints (detailed below).

SPIRAL constructs a roadmap to encode connectivity information about the samples in Ω. What is referred to as a roadmap in robot motion planning literature and in this paper is simply a nearest-neighbor graph $G = (V, E)$; V is the set of vertices, with each vertex corresponding to a conformation in the sampled ensemble Ω; E is the set of directed edges, with an edge connecting two vertices in V. E is initially populated with a set of pseudo-edges connecting nearest neighbors in Ω. A time-limited iterative interplay between a global search and local search/planners converts pseudo-edges residing in a user-specified number (K) of promising paths into tree search structures of actual edges. At the expiration of time or successful computation of K paths, the roadmap is augmented with conformations and

connections obtained by the local planners. All edge weights in the roadmap are recomputed to reflect energetic difficulty, and the resulting roadmap is queried for a specified number of lowest-cost paths. Various analyses can be conducted over the paths, in terms of energetics or proximity to the given start and goal conformations. We now describe each of the components in *SPIRAL*in greater detail.

A. Sampling Stage

SPIRAL extends the typical setting of a start and goal pair of conformations to an arbitrary number of conformations for which it will construct connecting paths. The idea is to accommodate applications where a number $\ell \geq 2$ of stable or semi-stable conformations are known from wet- or dry-laboratory studies for a protein of interest. The goal is to map out the connectivity among these conformations, highlighting how the protein can transition between any pair of these known conformations.

Let us refer to the input conformations as landmarks. The landmarks are used to initialize the Ω ensemble. The sampling stage consists of a cycle of selection and perturbation operations to populate Ω with more conformations. A conformation is first chosen from Ω by a selection operator. A perturbation operator is then sampled from a set of available ones and applied to the selected conformation to generate a new conformation. The generated conformation is checked for energetic feasibility prior to addition to the Ω ensemble. The process is repeated until $|\Omega|$ reaches a pre-determined value or the algorithm exhausts its computational budget.

1) Selection Operator: The selection operator is based on our prior work on tree-based methods for protein motion computation [21] but is extended here to deal with an arbitrary number of landmarks, arbitrary variables, and a more robust progress coordinate for conformation space coverage.

The goal of the selection operator is to promote coverage of the variable/conformation space enclosed by the landmarks. A progress coordinate, $\Delta R(C)_{i,j}$, is defined for each conformation C and a pair of landmarks (C_i, C_j) as in: $\Delta R(C)_{i,j} =$ lRMSD(C_i, C) $-$ lRMSD(C_j,C). lRMSD here refers to least root-mean-squared-deviation used to measure the dissimilarity between two conformations after optimal superposition removes differences due to rigid-body motions [30].

The $\Delta R(C)_{i,j}$ coordinate is used to guide sampling towards under-sampled regions of the conformation space as follows. For each pair of landmarks (C_i, C_j), a one-dimensional (1d) grid is defined over the range $[-\mathrm{lRMSD}(C_i, C_j) - 2, \mathrm{lRMSD}(C_i, C_j) + 2]$ rather than the range $[-\mathrm{lRMSD}(C_i, C_j), \mathrm{lRMSD}(C_i, C_j)]$. Since each cell in the grid is 1Å wide, the decision to extend the range by essentially two grid cells in each direction is so as to obtain a few more conformations. All conformations in Ω are projected onto the grid. In this way, each conformation in the growing ensemble Ω has $\binom{\ell}{2}$ projections, one in each of the $\binom{\ell}{2}$ grids.

The selection operator proceeds as follows. A pair of landmarks is first selected uniformly at random among the $\binom{\ell}{2}$ pairs. This determines the 1d grid, from which a cell is then sampled according to a probability distribution function defined by weights w_c associated with cells of the grid. To bias the selection of conformations from under-explored regions of the conformation space, w_c is set to $\frac{1}{(1+\mathrm{ns})*\mathrm{nc}}$, where ns is the number of times the

cell has been selected, and nc is the number of conformations projected onto that cell. Once a cell is selected, a conformation is then selected uniformly at random among conformations in the cell.

2) Perturbation Operators: In the absence of any specific insight onto the variables of relevance for a transition, *SPIRAL* employs a set of perturbation operators in order to make moves of different granularities in conformation space in the sampling stage. Each perturbation operator has to satisfy a set of energetic and geometric constraints.

One of the constraints enforces energetic feasibility of sampled conformations. The energy of a conformation $C^{'}$ generated from a selected conformation C, measured through the Rosetta *score3* function, is compared to the energy of C through the Metropolis criterion (*score3* is the backbone-level energy function, as we employ here only backbone-level representations of protein structure). If this fails, $C^{'}$ is not added to the ensemble. If it passes, $C^{'}$ is checked for satisfaction of distance-based constraints. It is worth noting that we have modified the score3 function to allow possible, partial unfolding of protein chains that may be crucial for a transition. This is done by setting the weight of the radius-of-gyration term to 0.0. This term in Rosetta penalizes non-compact conformations. By essentially removing it from the *score3* function, we allow a protein to elongate, or unfold, as needed in a transition.

Additional constraints are introduced on the minimum lRMSD ϵ_{min} of $C^{'}$ to any other conformation in the ensemble Ω and the maximum lRMSD δ of $C^{'}$ to the ℓ landmarks. The first constraint prevents redundant conformations from being added to Ω. The second constraint prevents sampling from veering off in regions of the conformation space deemed too far from the landmarks to be useful for participating in paths connecting them. While ϵ_{min} is a parameter that can depend on the specific system under investigation (analysis is provided in section III), a reasonable value for δ is 150% of the maximum lRMSD between any pairs of landmarks.

The idea behind making various perturbation operators available to *SPIRAL* is to allow *SPIRAL* to select the perturbation operator deemed most effective based on features of the conformation space and the specific problem at hand. For instance, when the goal is to connect landmarks that reside far away (several Å) in conformation space, a perturbation operator capable of making large moves is first desirable. Afterwards, to be able to make connections between such conformations, other perturbation operators capable of making smaller moves may be more effective.

We consider here three perturbation operators, detailed below. An optimal weighting scheme that is responsive to emerging features of the search space is difficult to formulate and beyond the scope of the work here. However, we have been able to empirically determine a weighting scheme that results in good baseline performance on all the protein systems studied here.

a) Molecular Fragment Replacement Operator: This operator is inspired from protein structure prediction, where backbone dihedral angles in a bundle/fragment of f consecutive amino acids are replaced altogether with values from a pre-compiled library. *SPIRAL* employs $f \in \{3, 9\}$ to balance between large ($f = 9$) and small ($f = 3$) moves.

b) Single Dihedral Replacement Operator: This operator modifies a single backbone dihedral angle at a time to allow small moves. From the selected conformation, a single dihedral angle is selected uniformly at random for perturbation. A new value is obtained by sampling from a normal distribution $\mathcal{N}(\mu, \sigma)$, where μ is the value of the

angle prior to perturbation. This operator, gaussian sampling, offers the option of biasing the selection of dihedral angles to promote selection of those that differ most between a selected conformation and a landmark, though our application of *SPIRAL* here does not make use of biased gaussian sampling in the sampling stage.

c) Reactive Temperature Scheme: The energetic constraint that determines whether a conformation C' produced from a perturbation operator applied onto a selected conformation C should be added to Ω is based on the Metropolis criterion. Essentially, a probability $e^{-(E_{C'}-E_C)/(K\cdot T)}$, is measured, where K is the Boltzmann constant, and T is temperature. An arbitrary temperature value is both difficult to justify and obtain constraint- satisfying conformations as Ω grows (if T is low). So, as in previous work on tree-based methods [21], we make use of a reactive temperature scheme but extend it to the multiple-landmark setting here. We maintain a temperature value T_c for each cell c of the 1d grids over the progress coordinate. Each cell's temperature is adjusted every s steps (typical value employed is 25). The temperature of a cell, T_c, is increased if the last s selections of that cell have resulted in no conformations being added to Ω. If conformations are added to Ω more than 60% of the time within a window of s steps, T_c is decreased. Increases and decreases occur over adjacent temperature levels per a proportional cooling scheme that starts with very high temperatures in the $2,000$K range and ends with room temperature of 300K.

B. Roadmap Building Stage

A two-layer, global-to-local, scheme is used, which is an iterative interplay between global and local search, summarized in pseudocode in Algorithm 1. First, all conformations in Ω are added to the vertex set V. For each $v \in V$, its k nearest-neighbors are identified, using lRMSD. For each identified neighbor, directional pseudo-edges are added. Additional pseudo-edges are added by identifying any vertex $< \epsilon_{max}$ from v that lies in a different connected component from v. Typical values for k and ϵ_{max} are 10 and 5Å, respectively. The pseudo-edges are assigned a weight to reflect their estimated difficulty of being realizable, as local planners have yet to be invoked. At initialization, all pseudo-edges are determined equally difficult and assigned the value 0.99 (line 2 in Algorithm 1). We note that this is an implementation detail, and other, positive values can be employed at the initialization stage.

Algorithm 1 The roadmap construction algorithm.

```
Input: G = (V, E)                                   ▷ The roadmap encoded as a graph
 1: for e ∈ E do
 2:     p(e) = 0.99
 3: end for
 4: while More Paths & Remaining Time do
 5:     P = CompLowCostPath(G)
 6:     for e ∈ P do
 7:         if e is unrealized then
 8:             p(e) = LocalPlanner(e, T)
 9:         end if
10:     end for
11: end while
```

The global search queries the roadmap for the current most promising/lowest-cost path in the roadmap connecting

two specified, start and goal conformations (line 5 in Algorithm 1). If there are unrealizable edges in the path, these edges are fed to the local search, which launches local planners on unrealized edges, pursuing path realization (lines 6-10 in Algorithm 1). The local planners are given a limited computational budget, and they report at the end of this budget either a realized edge or a new weight for the unrealized edges (line 8 in Algorithm 1).

In this iterative interplay between path query and path realization, over time, the pseudo-edges that are most difficult to realize will be assigned high weights and will thus be unlikely to participate in the lowest-cost path pushed to the local planners. This dynamic interplay apportions computational resources in a manner that promotes rapid path discovery. The iterative process continues until a total computational budget is exhausted or a user-specified number of paths is obtained.

We now proceed to provide more details into the path query and realization interplay. A pair of landmarks are selected uniformly at random over the $\ell!$ permutations. The roadmap is then queried for a lowest-cost path, using the assigned pseudo-edge weights. Yen's K-Shortest path algorithm [31] is used to identify the lowest non-zero cost path in the roadmap and allow obtaining paths after the first one has been realized. Given an identified path, a local planner is assigned to any of the unrealized edges. The planner is given a fixed computational budget, time T. If the local planner succeeds, the pseudo-edge it has realized is assigned a weight of 0 to indicate the pseudo-edge is resolved. If the local planner fails, the pseudo-edge is reweighted: $w_e = 0.7 \cdot \mathrm{CallsToPlanner} + 0.3 \cdot (\mathrm{ClosestNode} - \mathrm{RequireResolution})^2$. $\mathrm{CallsToPlanner}$ tracks the number of times the planner has been requested to work on a particular pseudo-edge, $\mathrm{ClosestNode}$ is the node in the tree constructed by the local planner that is closest to the vertex v in the directed pseudo-edge (u, v). For the planner to be successful, it must also generate a path that is within a user-specified lRMSD of the vertex v.

SPIRAL learns from the feedback it receives from local planners. When a local planner has failed to complete a path more than a set number of times, *SPIRAL* augments the graph with conformations identified by the local planner that are otherwise invisible to the global layer. The intuition behind this strategy is to identify difficult regions for making connections that may benefit from further conformation sampling. We now proceed to relate details on the local planner and the augmentation procedure.

d) Local Planner: The local planner is an adaption of the tree-based method proposed in [21]. The adaptation consists of diversifying the types of perturbation operators employed in the expansion of the tree. The local planner selects through a probabilistic scheme shown in section III from the menu of perturbation operators described above. While biased gaussian sampling is not used in the sampling stage in *SPIRAL*, it is used by the local planner.

e) Roadmap Augmentation: Some regions of the conformation space may be particularly challenging to connect through local planners. This can be due to high energetic barriers or inadequate sampling. To address this issue, *SPIRAL* makes use of a feedback mechanism to augment the roadmap. When a local planner encounters difficulty realizing a pseudo-edge connecting given conformations p and r more than RefineLimit times (set at 25), the problem of connecting p to r is considered as a mini-version of the entire transition path sampling problem. The sampling scheme is repeated, essentially treating p and r as start and goal conformations. The perturbation operators described above are used together with a new one based on straight-line interpolation. The produced conformations

are then minimized using the Rosetta *relax* protocol. Only the lowest-energy conformation is considered for addition. Conformations obtained from the perturbation operators are checked for satisfaction of the energetic and geometric constraints also used in the sampling stage. The operators are applied under a probabilistic scheme detailed in section III until either 25 conformations have been added to the roadmap or a maximum of 2500 attempts have been made.

f) Roadmap Analysis: Each edge in the roadmap is reweighted per the Metropolis criterion to reflect the energetic difficulty. A room temperature value is used for this purpose. The reweighted graph is queried for one or more lowest-cost paths, which are then analyzed in terms of energetic profile or distance within which they come of the goals, as related in section III.

III. Results

A. Test Systems

We evaluate *SPIRAL* on 9 protein systems which are listed in Table I. These systems were selected to allow comparisons to other published studies. We construct paths between given, known conformations for each system, and detail the lRMSD distance between these states in Table I. We have observed that the lRMSD distance between functional states is not by itself a good indicator of system difficultly. Larger proteins may require some unfolding to allow for collision free motions between closed and open structural states. This transition can also involve crossing large energetic barriers, which is why some computational studies focus on open to closed transitions. In this work, we consider transitions in both directions.

TABLE I
PROTEIN SYSTEMS FOR EVALUATION OF PERFORMANCE.

System	Length	Start ↔ Goal	lRMSD(start, goal)
CVN	101	2ezm ↔ 1l5e	16.0 Å
CaM	140	1cfd ↔ 1cll	10.7 Å
		1cfd ↔ 2f3y	9.9 Å
		1cfd ↔ 1lin	10.0 Å
		1cll ↔ 2f3y	13.4 Å
		1cll ↔ 1lin	15.0 Å
		2f3y ↔ 1lin	4.3 Å
AdK	214	1ake ↔ 4ake	7.0 Å
LAO	238	1laf ↔ 2lao	4.7 Å
URP	271	1urp ↔ 2dri	4.1 Å
DAP	320	1dap ↔ 3dap	4.3 Å
OMP	370	1omp ↔ 3mbp	3.7 Å
NS3	436	3kqk ↔ 3kql	3.5 Å
BKA	691	1cb6 ↔ 1bka	6.4 Å

B. Implementation Details

SPIRAL is implemented in C++. For each protein, the sampling stage attempts to generate $10,000$ samples. If the total number of energy evaluations exceeds $1,000$ times the requested ensemble size, the sampling stage is

terminated. That is, a maximum of $1,000$ attempts are made to obtain each sample. The roadmap building stage is terminated after $10,000$ iterations of the interplay between path query and path realization. This stage may terminate earlier if $K = 250$ paths are obtained for each of the $\ell * (\ell - 1)$ transitions as a way to control computational cost. The analysis stage reports the 50 lowest-cost paths. In terms of CPU time, the computational time demands of all three stages in *SPIRAL* spans anywhere from 56 hours on one CPU for protein systems around 100 amino acids long to 300 hours on one CPU for systems around 700 amino acids long.

During the sampling stage, the molecular fragment replacement perturbation operator with $f = 3$ is selected 75% of the time, the same operator with $f = 9$ is selected 20% of the time, and the gaussian sampling operator is selected 5% of the time. The reason for this scheme is to make large moves more often than small ones so as to spread out conformations in conformation space during sampling.

During the roadmap building stage, the probabilistic scheme with which local planners and the roadmap augmentation make use of the perturbation operators is different, as shown in Table II. A local planner can use two different schemes depending on the lRMSD between the two conformation/vertices it is asked to connect by the global layer. These schemes are not fine-tuned; essentially, when the distance is ≤ 2.5Å, smaller moves are promoted as opposed to when the distance is > 2.5Å. The reason for basing the decision at 2.5Å is due to prior work on tree-based planners showing that molecular fragment replacement can result in step sizes greater than 2.5Å [21].

TABLE II
THE PERTURBATION OPERATOR SET AND THEIR WEIGHTS DURING ROADMAP CONSTRUCTION. GAUSSIAN OPERATIONS ALL HAVE THEIR MEAN CENTERED AT ZERO.

Process	Perturbation Operator	Prob.
Local Planner ($>$ 2.5 Å lRMSD)	Fragments ($f = 3$)	0.70
	Gaus ($\sigma = 15$)	0.15
	Biased Gaus ($\sigma = 15$)	0.15
Local Planner ($\leq$ 2.5 Å lRMSD)	Fragments ($f = 3$)	0.20
	Gaus (σ = 15)	0.40
	Biased Gaus (σ = 15)	0.40
Augmentation	Fragments($f = 3$)	0.20
	Gaus (σ = 15)	0.40
	Biased Gaus (σ = 15)	0.40
	Interpolation-based	0.05

The ϵ_{min} parameter controls how close neighboring conformations will be in the roadmap. Intuitively, smaller ϵ_{min} values would produce a better-quality roadmap. Our analysis indicates that this is not the case. Small values of ϵ_{min} (< 1Å) can result in many small cliques being formed in the roadmap around local minima conformations. This is not surprising, particularly for the broad minima that contain the stable and semi-stable landmarks. For these minima, it is rather easy to sample a very large number of conformations nearby a landmark and thus essentially "get stuck" in the same local minimum. Insisting on a minimum distance separation among sampled conformations forces sampling not to provide refinement or exploitation of a particular local minimum but rather explore the breadth

of the conformation space. Not insisting on a minimum distance pushes all the work to obtaining intermediate conformations to bridge local minima to the local planners, which is an ineffective use of computational time. The ϵ_{min} parameter is set to 2.0Å for systems where the lRMSD between landmarks is > 6Å, 1.5Å for systems where the lRMSD between landmarks is > 4.5 but ≤ 6Å, and 1.0 for systems where the lRMSD between landmarks is ≤ 4.5Å.

C. Comparison of Paths with Other Methods

We compare the paths obtained with *SPIRAL* to other published tree-based methods [21], [20], [16], [17]. These other methods make use of protein specific attributes to customize their perturbation operations/move sets. For instance, our tree-based method in [21] uses molecular fragment replacements with $f = 3$, the method in [20] uses moves over low-frequency modes revealed by normal mode analysis, and the method in [16], [17] considers only backbone dihedral angles whose values change between the given functional conformations. The last two methods consider a low-dimensional search space of no more than 30 dimensions.

We report the closest that any path computed by *SPIRAL* comes to the specified goal conformation and compare such values on all protein systems to those reported in other published work, which is showcased in Figure 3 and detailed in Table III. Columns $4-7$ in Table III show these values for *SPIRAL* and other published work. Column 3 reports some more details on the path with which *SPIRAL* comes closest to the goal conformation by listing the maximum lRMSD between any two consecutive conformations in the path. *SPIRAL* typically generates paths with conformations closer to the goal conformation than other methods (highlighted in bold where true). A video illustrating the lowest-cost conformational path reported by *SPIRAL* for the CVN protein can be found at http://youtu.be/7P4reYO3k-c.

D. Analysis of Energetic Profiles

We show the energetic profile of the lowest-cost path obtained by *SPIRAL* on the AdK system. We compare these profiles to those obtained by the interpolation-based planner described in section II. For this planner, the resolution distance ϵ is set to 1.0 Å, and 50 cycles are performed to obtain a path. This provides a fair comparison, given that we also analyze 50 paths obtained after the analysis stage in *SPIRAL* and report here the lowest-cost one. Figure 1 shows that on proteins, such as AdK, where the distance between the start and goal conformations is large, paths provided by the interpolation-based planner tend to have higher energies than those provided by *SPIRAL*.

E. Path Variance

SPIRAL discovers multiple paths between landmark conformations. We illustrate here the diversity of paths on two different systems, AdK and CVN as follows. Paths are visualized in a 2d embedding, projecting each conformation in a path onto a 2d grid that tracks the lRMSD of a conformation from the start (horizontal axis) and goal (vertical axis) conformations, respectively. The grid is color-coded in a grayscale scheme, with darker colors showing more paths going through a particular cell of the grid. The result of this visualization technique is a heatmap that illustrates path diversity. Results for two different start and goal pairs for AdK and CVN systems are shown in Figure 2.

TABLE III

COLUMN 4 REPORTS SMALLEST DISTANCE TO GOAL OVER ALL PATHS OBTAINED BY *SPIRAL*. COLUMNS $5-7$ SHOWS SUCH DISTANCES FROM TREE-BASED METHODS (EST [21], RRT CORTÉS[20], AND PDST HASPEL [16], [17]). MAX STEP IN COLUMN 3 REFERS TO THE MAXIMUM LRMSD BETWEEN ANY TWO CONSECUTIVE CONFORMATIONS IN THE *SPIRAL* PATH THAT COMES CLOSEST TO THE GOAL. '-' INDICATES LACK OF PUBLISHED DATA.

Syst	Start → Goal	Max Step	Dist to Goal (Å)			
			Spiral	EST	RRT	PDST
CVN 101	2ezm → 1l5e	1.5	**1.5**	–	2.1	2.1
	1l5e → 2ezm	1.5	1.3	–	–	–
CaM 144	1cll → 1cfd	2.0	**1.5**	3.35	–	–
	1cfd → 1cll	2.0	**0.6**	3.17	–	–
	1cll → 2f3y	2.0	**0.9**	1.67	–	–
	2f3y → 1cll	2.0	**0.6**	0.73	–	1.33
	1cfd → 2f3y	2.0	**0.9**	3.5	–	–
	2f3y → 1cfd	2.0	**1.46**	3.2	–	–
	1cll → 1lin	2.0	**2.0**	–	–	–
	1lin → 1cll	2.0	**0.6**	–	–	–
	1cfd → 1lin	2.0	**2.0**	–	–	–
	1lin → 1cfd	2.0	**1.8**	–	–	–
	2f3y → 1lin	2.0	**2.0**	–	–	–
	1lin → 2f3y	2.0	**0.9**	–	–	–
AdK 214	1ake → 4ake	3.0	**1.86**	3.8	2.56	2.2
	4ake → 1ake	3.12	**1.33**	3.6	1.56	–
Lao 238	2lao → 1laf	2.0	**1.21**	–	1.32	–
	1laf → 2lao	3.2	1.90	–	–	–
URP 271	1urp → 2dri	NA	**1.21**	–	1.32	–
	2dri → 1urp	NA	1.90	–	–	–
DAP 320	1dap → 3dap	1.42	1.5	–	1.31	–
	3dap → 1dap	1.46	0.92	–	–	–
OMP 370	1omp → 3mbp	1.04	3.04	–	–	–
	3mbp → 1omp	0.91	3.61	–	–	–
NS3 436	3kqk → 3kql	1.9	**0.9**	–	1.3	–
	3kql → 3kqk	1.9	**1.0**	–	–	–
BKA 691	1bka → 1cb6	3.87	**1.55**	–	2.79	–
	1cb6 → 1bka	3.98	1.69	–	–	–

We draw attention to the heatmaps for AdK. Comparisons of these heatmaps with the energy profiles drawn in Figure 1 show that the highest concentration of conformations needed for the transitions is on regions of high energies. This illustrates that transitions in AdK go through high-energy barriers and that *SPIRAL* discovers different ways of crossing the barriers.

IV. CONCLUSIONS

This paper has proposed *SPIRAL*, a novel transition path sampling algorithm capable of handling proteins of various sizes and settings where distances among conformations of interest can exceed 16Å. The algorithm is inspired by popular frameworks in robot motion planning as opposed to MD- or MC-based frameworks. The main

reason for pursuing a robotics-inspired treatment is to address the issue of insufficient sampling in MD- or MC-based frameworks, particularly when transitions involve disparate time and length scales.

However, it is worth emphasizing that robotics-inspired methods have yet to become standard tools via which researchers can study protein equilibrium dynamics. The very techniques employed by these methods to address outstanding challenges in sampling may also result in paths that do not faithfully capture probable transitions. One of the reasons for the proposed *SPIRAL* in this paper is to provide a first-generation, general algorithm to spur further research on robotics-inspired treatments of protein structural transitions.

SPIRAL exploits no particular information for a given protein, which allows broad applicability. It is expected that tunings of the probabilistic scheme or employment of additional perturbation operators and moves based on system-specific insight will improve performance. Future work will consider such directions, but the current need of the community is for a powerful, general, baseline method for the purpose of benchmarking.

The results shown here suggest *SPIRAL* produces good-quality paths and can be employed both to extract information on protein transitions, possible long-lived intermediate conformations in such transitions, as well as to advance algorithmic work in transition path sampling. In particular, the inherent prioritization scheme in *SPIRAL* allows the sampling of both low-cost paths and high-cost paths, provided enough computational budget is allocated. The latter paths may highlight possible local unfolding involved in protein motions connecting functional conformations. An executable of *SPIRAL* can be provided to researchers upon demand.

ACKNOWLEDGEMENT

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. 1440581 and CAREER Award No. 1144106) and the Kate Miller Jeffress Memorial Trust Award. Computations 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).

REFERENCES

[1] K. Jenzler-Wildman and D. Kern, "Dynamic personalities of proteins," *Nature*, vol. 450, pp. 964–972, 2007.

[2] R. P. Feynman, R. B. Leighton, and M. Sands, *The Feynman Lectures on Physics*. Reading, MA: Addison-Wesley, 1963.

[3] A. Cooper, "Protein fluctuations and the thermodynamic uncertainty principle," *Prog Biophys Mol Biol*, vol. 44, no. 3, pp. 181–214, 1984.

[4] H. Frauenfelder, S. G. Sligar, and P. G. Wolynes, "The energy landscapes and motion on proteins," *Science*, vol. 254, no. 5038, pp. 1598–1603, 1991.

[5] M. Vendruscolo and C. M. Dobson, "Dynamic visions of enzymatic reactions," *Science*, vol. 313, no. 5793, pp. 1586–1587, 2006.

[6] J. S. Hub and B. L. de Groot, "Detection of functional modes in protein dynamics," *PLoS Comp Biol*, vol. 5, no. 8, p. e1000480, 2009.

[7] D. D. Boehr, R. Nussinov, and P. E. Wright, "The role of dynamic conformational ensembles in biomolecular recognition," *Nature Chem Biol*, vol. 5, no. 11, pp. 789–96, 2009.

[8] I. Bahar, T. R. Lezon, L. W. Yang, and E. Eyal, "Global dynamics of proteins: bridging between structure and function," *Annu Rev Biophys*, vol. 39, pp. 23–42, 2010.

[9] C. F. Wong and M. J. A., "Protein simulation and drug design," *Adv. Protein Chem.*, vol. 66, no. 1, pp. 87–121, 2003.

[10] M. Merkx, M. V. Golynskiy, L. H. Lindenburg, and J. L. Vinkenborg, "Rational design of FRET sensor proteins based on mutually exclusive domain interactions," *Biochem Soc Trans*, vol. 41, no. 5, pp. 128–134, 2013.

[11] K. Okazaki, N. Koga, S. Takada, J. N. Onuchic, and P. G. Wolynes, "Multiple-basin energy landscapes for large-amplitude conformational motions of proteins: Structure-based molecular dynamics simulations," *Proc. Natl. Acad. Sci. USA*, vol. 103, no. 32, pp. 11 844–11 849, 2006.

[12] H. M. Lee, K. S. M., H. M. Kim, and Y. D. Suh, "Single-molecule surface-enhanced Raman spectroscopy: a perspective on the current status," *Phys Chem Chem Phys*, vol. 15, pp. 5276–5287, 2013.

[13] R. E. Amaro and M. Bansai, "Editorial overview: Theory and simulation: Tools for solving the insolvable," *Curr. Opinion Struct. Biol.*, vol. 25, pp. 4–5, 2014.

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

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

[16] N. Haspel, M. Moll, M. L. Baker, W. Chiu, and K. L. E., "Tracing conformational changes in proteins," *BMC Struct. Biol.*, vol. 10, no. Suppl1, p. S1, 2010.

[17] D. Luo and N. Haspel, "Multi-resolution rigidity-based sampling of protein conformational paths," in *CSBW (Computational Structural Bioinformatics Workshop), in proc. of ACM-BCB (ACM International conference on Bioinformatics and Computational Biology)*, September 2013, pp. 787–793.

[18] J. Cortés, T. Simeon, R. de Angulo, D. Guieysse, M. Remaud-Simeon, and V. Tran, "A path planning approach for computing large-amplitude motions of flexible molecules," *Bioinformatics*, vol. 21, no. S1, pp. 116–125, 2005.

[19] L. Jaillet, F. J. Corcho, J.-J. Perez, and J. Cortés, "Randomized tree construction algorithm to explore energy landscapes," *J. Comput. Chem.*, vol. 32, no. 16, pp. 3464–3474, 2011.

[20] I. Al-Bluwi, M. Vaisset, T. Siméon, and J. Cortés, "Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods," *BMC Structural Biology*, vol. 13, no. Suppl 1, p. S8, 2013.

[21] K. Molloy and A. Shehu, "Elucidating the ensemble of functionally-relevant transitions in protein systems with a robotics-inspired method," *BMC Struct Biol*, vol. 13, no. Suppl 1, p. S8, 2013.

[22] D. Devaurs, K. Molloy, M. Vaisset, A. Shehu, T. Siméon, and C. Cortés, "Characterizing energy landscapes of peptides using a combination of stochastic algorithms," *IEEE Trans Nanobioscience*, vol. 14, no. 5, pp. 1–8, 2015.

[23] 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.*, vol. 12, no. 4, pp. 566–580, 1996.

[24] N. M. Amato, K. A. Dill, and G. Song, "Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures," *J. Comp. Biol.*, vol. 10, no. 3-4, pp. 239–255, 2002.

[25] S. Thomas, G. Song, and N. Amato, "Protein folding by motion planning," *Physical Biology*, no. 2, pp. S148–S155, 2005.

[26] S. Thomas, X. Tang, L. Tapia, and N. M. Amato, "Simulating protein motions with rigidity analysis," *J. Comput. Biol.*, vol. 14, no. 6, pp. 839–855, 2007.

[27] L. Tapia, X. Tang, S. Thomas, and N. Amato, "Kinetics analysis methods for approximate folding landscapes," *Bioinformatics*, vol. 23, p. i539i548, 2007.

[28] L. Tapia, S. Thomas, and N. Amato, "A motion planning approach to studying molecular motions," *Communications in Information Systems*, vol. 10, no. 1, pp. 53–68, 2010.

[29] C. Nielsen and L. Kavraki, "A two level fuzzy prm for manipulation planning," in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 3, 2000, pp. 1716–1721 vol.3.

[30] A. D. McLachlan, "A mathematical procedure for superimposing atomic coordinates of proteins," *Acta Crystallogr. A.*, vol. 26, no. 6, pp. 656–657, 1972.

[31] Y. JY, "Finding the k shortest loop less paths in a network." *Management Science*, vol. 17, pp. 712–716, 1971.

Kevin Molloy Dr. Kevin Molloy received his B.S., M. S. and Ph.D. in Computer Science from George Mason University in 1998, 2011, and 2015. He is currently working as a postdoctoral fellow at CNRS-LAAS in Toulouse, France where he is researching computational methods for modeling protein dynamics and motion.

Amarda Shehu Dr. Amarda Shehu is an Associate Professor in the Department of Computer Science at George Mason University. She holds affiliated appointments in the Department of Bioengineering and School of Systems Biology at George Mason University. She received her B.S. in Computer Science and Mathematics from Clarkson University in Potsdam, NY and her Ph.D. in Computer Science from Rice University in Houston, TX, where she was an NIH fellow of the Nanobiology Training Program of the Gulf Coast Consortia. Shehu's research contributions are in computational structural biology, biophysics, and bioinformatics with a focus on issues concerning the relationship between sequence, structure, dynamics, and function in biological molecules. Her research on probabilistic search and optimization algorithms for protein structure modeling is supported by various NSF programs, including Intelligent Information Systems, Computing Core Foundations, and Software Infrastructure. Shehu is also the recipient of an NSF CAREER award in 2012. She is a member of the IEEE and ACM.

FIGURE CAPTIONS

Figure 1

Top panel: Each conformation generated in the sampling stage for AdK is projected onto a 2d heatmap based on its lRMSD distance to the start (horizontal axis) and goal (vertical axis) conformation, respectively. The blue line shows the direct interpolation path between the start and the goal. The positions of the *SPIRAL* (triangles) and the interpolated-based planner (squares) are shown. Bottom panel: The corresponding energy profiles of the paths from each planner are shown.

Figure 2

SPIRAL computed paths for the transition Adk (left panels) and CVN (right panels) are visualized here by projecting each conformation in a path onto a 2d grid; coordinates of the grid track lRMSD from the start (horizontal axis) and the goal (vertical axis) conformations. The blue line shows the direct line interpolation between the start and goal conformations. Cells of the grid are color-coded, with darker colors indicating more paths going through the particular cell of the grid.

Figure 3

SPIRAL is compared to other methods. The lowest lRMSD with which a *SPIRAL* path comes to the goal conformation is compared to the lowest lRMSD from other methods. The comparison is limited to the best method (achieving lowest lRMSD to the goal) on the particular protein at hand.

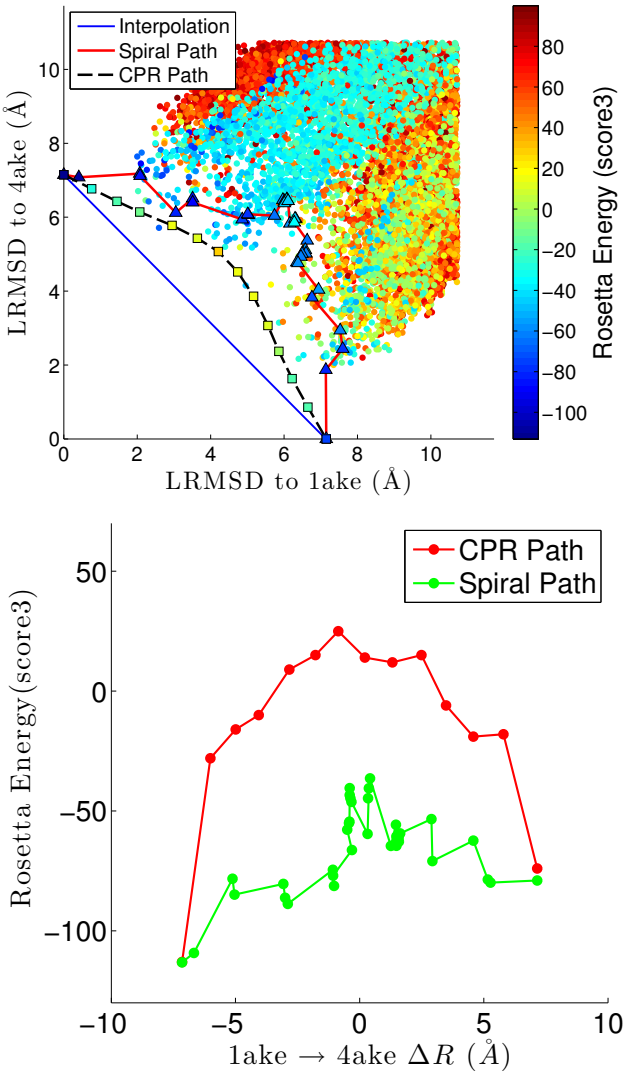
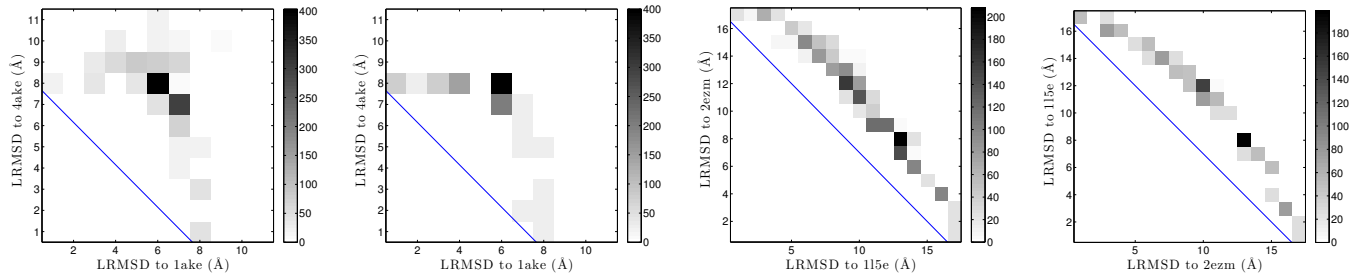

Interpolation
Spiral Path
CPR Path
LRMSD to 4ake (Å)
LRMSD to 1ake (Å)
Rosetta Energy (score3)
CPR Path
Spiral Path
Rosetta Energy (score3)
1ake → 4ake ΔR (Å)

Fig. 1.



1ake $\rightarrow$ 4ake 4ake $\rightarrow$1ake 1l5e $\rightarrow$ 2ezm 2ezm $\rightarrow$ 1l5e

Fig. 2.

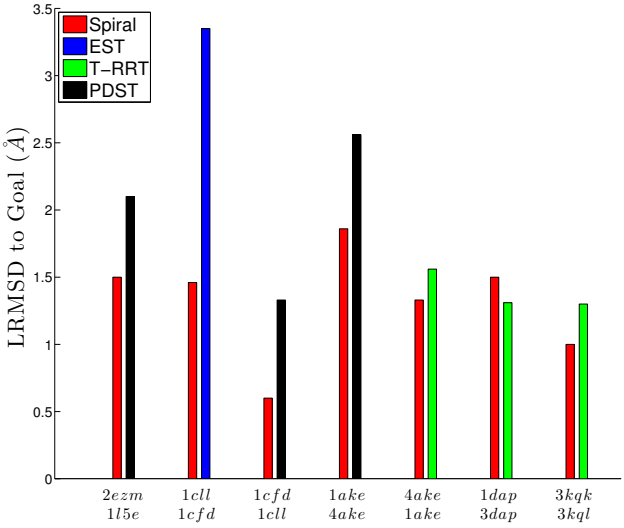

Spiral
EST
T−RRT
PDST
LRMSD to Goal ($\mathring{A}$)
3.5
3
2.5
2
1.5
1
0.5
0
$2ezm$ $1l5e$
$1cll$ $1cfd$
$1cfd$ $1cll$
$1ake$ $4ake$
$4ake$ $1ake$
$1dap$ $3dap$
$3kqk$ $3kql$

Fig. 3.