JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

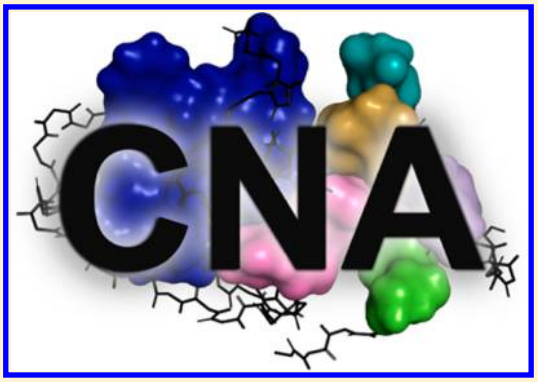
pubs.acs.org/jcim

Constraint Network Analysis (CNA): A Python Software Package for Efficiently Linking Biomacromolecular Structure, Flexibility, (Thermo-)Stability, and Function

Christopher Pfleger,[‡] Prakash Chandra Rathi,[‡] Doris L. Klein, Sebastian Radestock,[†] and Holger Gohlke*

Institute for Pharmaceutical and Medicinal Chemistry, Department of Mathematics and Natural Sciences, Heinrich-Heine-University, Universitätsstr. 1, 40225, Düsseldorf, Germany

ABSTRACT: For deriving maximal advantage from information on biomacromolecular flexibility and rigidity, results from rigidity analyses must be linked to biologically relevant characteristics of a structure. Here, we describe the Python-based software package Constraint Network Analysis (CNA) developed for this task. CNA functions as a front- and backend to the graph-based rigidity analysis software FIRST. CNA goes beyond the mere identification of flexible and rigid regions in a biomacromolecule in that it (I) provides a refined modeling of thermal unfolding simulations that also considers the temperature-dependence of hydrophobic tethers, (II) allows performing rigidity analyses on ensembles of network topologies, either generated from structural ensembles or by using the concept of fuzzy noncovalent constraints, and (III) computes a set of global and local indices for quantifying biomacromolecular stability. This leads to more robust results from rigidity analyses and extends the application domain of rigidity analyses in that phase transition points ("melting points") and unfolding nuclei ("structural weak spots") are determined automatically. Furthermore, CNA robustly handles small-molecule ligands in general. Such advancements are important for applying rigidity analysis to data-driven protein engineering and for estimating the influence of ligand molecules on biomacromolecular stability. CNA maintains the efficiency of FIRST such that the analysis of a single protein structure takes a few seconds for systems of several hundred residues on a single core. These features make CNA an interesting tool for linking biomacromolecular structure, flexibility, (thermo-)stability, and function. CNA is available from http://cpclab.uni-duesseldorf.de/software for nonprofit organizations.



INTRODUCTION

The concepts of biomacromolecular flexibility and its opposite, rigidity, are crucial for understanding the relationship between biomacromolecular structure, (thermo-)stability, and function. In the field of statics, flexibility and rigidity denote the possibility (or impossibility) of internal motion but are not associated with information about directions and magnitudes of movements. Identifying and modulating the heterogeneous composition of biomacromolecules in terms of flexible and rigid regions is becoming increasingly important for successful protein engineering and rational drug-design.[1−5] Several computational approaches have been developed that identify flexible and rigid regions by either determining spatial variations in the local packing density[6] or representing and analyzing a structure as a connectivity network of interacting atoms or residues.[7−12] The approaches benefit from being computationally highly efficient. A related concept has been introduced by Jacobs et al.[13] Here, biomacromolecules were initially represented as bond-bending networks in which each atom has three degrees of freedom representing the dimensions of motion in 3-space. In later versions, the equivalent body-bar representation is used where atoms are modeled as bodies with six degrees of freedom.[13−15] By adding constraints (representing covalent and noncovalent bonds in a biomacromolecular context) between the bodies, internal motions become restricted. Each constraint is modeled as a set of bars, and each bar removes one degree of freedom. According to the type of interaction, the number of bars varies in that stronger interactions are modeled with a higher number of bars than weaker ones. Noncovalent interactions such as hydrogen bonds, salt bridges, hydrophobic tethers, and stacking interactions contribute most to the biomacromolecular stability; hence, these interactions are modeled as constraints in addition to covalent bonds. Once the network is constructed, the Pebble Game algorithm, available within the FIRST (Floppy Inclusions and Rigid Substructure Topography) software, efficiently decomposes the network into rigid clusters and flexible hinge regions from the number and spatial distribution of bond-rotational degrees of freedom.[16,17] A rigid region is a collection of interlocked bonds allowing no relative motion of the bodies. Such a region can either be overconstrained, if it has redundant constraints, or is isostatically rigid. In a flexible region, dihedral rotation is not locked in by other bonds. The theory underlying this approach is rigorous[18] and has been applied in different areas of biomacromolecular research.[5,19−35]

Received: January 20, 2013
Published: March 21, 2013

We developed the command-line Python-based software package Constraint Network Analysis (CNA) for analyzing structural features of biomacromolecules that are important for the molecule's stability. CNA functions as a front- and backend to the FIRST software and allows (I) setting up a variety of constraint network representations for analysis by FIRST, (II) processing the results obtained from FIRST, and (III) calculating seven indices for quantifying biomacromolecular stability, both globally and locally.[36] As to the latter, the indices are calculated by monitoring changes of the network stability along a thermal unfolding simulation. The thermal unfolding is simulated by consecutively removing hydrogen bond (including salt bridge) constraints from the network with increasing temperature. Thermal unfolding simulations have been successfully applied in several studies on proteins, RNAs, and the ribosome in order to understand how flexibility and rigidity is linked to biomacromolecular stability and function.[4,5,14,19,28,31,34,35,37]

CNA goes beyond the mere identification of flexible and rigid regions in a biomacromolecular structure in that it allows linking results from constraint network analysis to biologically relevant characteristics of a structure. This is key for deriving maximal advantage from information on biomacromolecular flexibility and rigidity. Here, we describe the design and implementation of the CNA software package. We then demonstrate its application scope in a showcase example on Hen Egg White Lysozyme (HEWL) structures. The CNA software package is available under an academic license from http://cpclab.uni-duesseldorf.de/software.

METHODS AND IMPLEMENTATION

General Overview. The CNA software package allows three different types of rigidity analysis: (I) based on a *single network topology* generated from a single input structure, (II) based on an *ensemble of network topologies* generated from a conformational ensemble provided as input,[21,35] and (III) based on an *ensemble of network topologies* generated from a single input structure by considering fuzzy noncovalent constraints (FNC) (C. Pfleger, H. Gohlke, to be published elsewhere). The last variant mimics that noncovalent constraints thermally break and reform even in the native state of a biomacromolecule.[38] In short, we developed a system-independent parametrization of fuzzy noncovalent constraints by analyzing the atom type and location-dependent persistence characteristics of noncovalent constraints (hydrogen bonds, salt-bridges, and hydrophobic tethers) during MD simulations. With this, the number and distribution of noncovalent constraints are modulated by random components within certain ranges, simulating thermal fluctuations of a biomacromolecule without actually moving atoms. In the related distance constraint model (DCM), ensembles of network topologies are generated considering mean-field probabilities of hydrogen bond and torsion constraints in a Monte Carlo sampling.[20,39] Average stability characteristics are then calculated by constraint counting on each topology in the ensemble.[40] As a downside, the DCM approach requires experimental data for a system-specific parametrization of the model.

The analysis of a *single network topology* by CNA consists of the following steps. Initially, a constraint network is generated from the input structure by placing covalent and noncovalent constraints according to rules described in refs 13−15. Next, a thermal unfolding simulation is carried out by sequentially removing noncovalent constraints from the network (see section Thermal Unfolding Simulation for details). For each network during the simulation, a rigidity analysis by FIRST is performed and then post-processed to calculate global and local indices to characterize biomacromolecular flexibility and rigidity. The workflow of the software is illustrated in Figure 1. In the case of analyzing an *ensemble of network topologies*, these steps are repeated for each network, and the results are averaged over the ensemble.

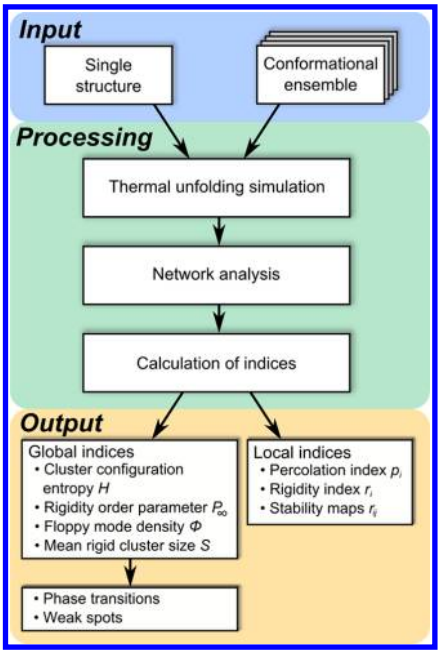


Figure 1. Schematic workflow of the CNA software.

Upon running a thermal unfolding simulation (a) phase transition(s) can be identified at which the network changes from mainly rigid to flexible. For this, the change in the global indices is monitored during the simulation. Four different global indices are implemented in CNA. They monitor (I) the normalized number of independent internal degrees of freedom (floppy mode density, Φ), (II) the fraction of the network belonging to a rigid component (rigidity order parameter, P_∞), (III) the degree of disorder in the network (cluster configuration entropy, H), and the rigid cluster size distribution (mean rigid cluster size, S). In addition, CNA calculates three local indices that characterize the flexibility and rigidity at the bond level: (I) the percolation index p_i monitors the percolation behavior of a biomacromolecule on a microscopic level and thus allows the identification of the hierarchical organization of the giant percolating cluster during a thermal unfolding simulation, (II) the rigidity index r_i monitors when a bond segregates from a rigid cluster, (III) a stability map is a two-dimensional itemization of the rigidity index r_i and is derived by identifying "rigid contacts" between two residues. Exact definitions of these indices and guidelines for when to use them are given in ref 36. Furthermore, the CNA software identifies unfolding nuclei, i.e., those residues that break apart from the giant cluster at the phase transition point.[4,28,35] The unfolding nuclei can be considered weak spots in the structure; accordingly, this knowledge can be exploited in data-driven protein engineering to focus on residues that are highly likely to improve thermostability upon mutation.

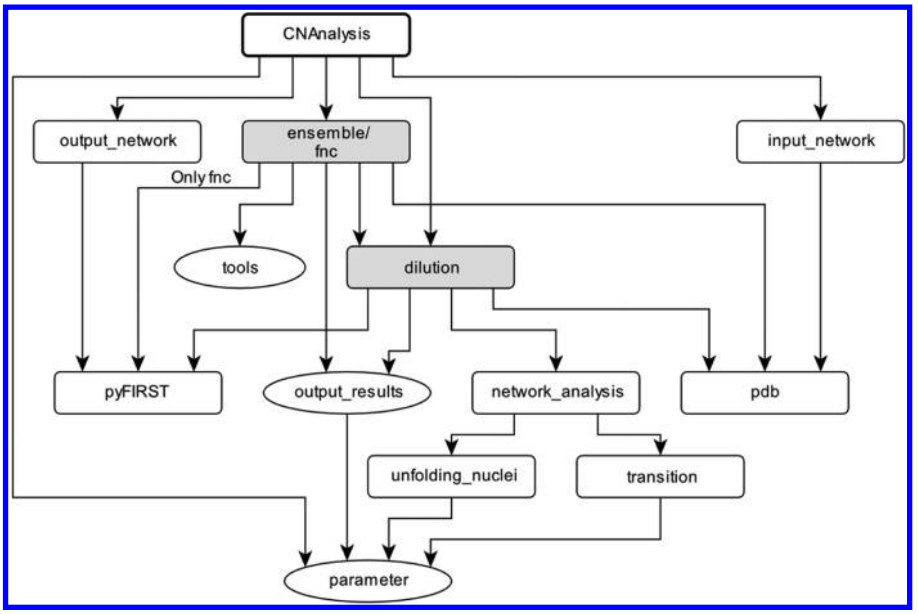


Figure 2. Hierarchical structure of the CNA software. All modules that contain (a) class definition(s) are shown in rectangles. The core module *CNAnalysis* is highlighted by a bold frame. Modules colored in gray contain the simulation methods for analyzing a single network topology and an ensemble of network topologies. Modules that solely contain methods are shown as ellipses. An arrow indicates the call of a module by another module.

CNA is implemented as a Python-based software package making use of an object-oriented design (Figure 2). Third party software is required for full functionality (Table 1): (I) The Biopython package[41] is needed to parse input PDB files and provides information on secondary structure from a DSSP analysis.[42,43] (II) For statistical analysis and detecting the phase transitions, the Numpy[44] and SciPy[45] extensions for Python are required. (III) The Open Babel[46,47] Python-bindings are required to determine the bond order of small-molecule ligands. To facilitate the installation of the CNA software package, the third party software is provided with the CNA source tree except for the DSSP program, which is available at http://swift.cmbi.ru.nl/gv/dssp/. The CNA source tree also contains a comprehensive documentation detailing the installation and usage of the software and a suite of test cases to check the validity of the installation. CNA is a command-line based software that is called by the shell script *CNA.sh*. A "--help" argument lists all available options and required arguments, their descriptions, default values, and the range of allowed values. An erroneous argument set for an option produces an informative error message. The CNA software has been successfully tested on Debian, OpenSuse, and CentOS Linux platforms.

Table 1. External Software Needed by the CNA Software

name	version	description and use
Python	2.73	Python interpreter used by the CNA software package
Biopython	1.58	for reading PDB files using the Bio.PDB package and parsing results from the DSSP program using the Bio.DSSP package
NumPy	1.6.1	for statistical analyses
SciPy	0.11.0	for statistical analyses
Open Babel	2.3.1	for identifying the connectivity and bond orders of ligand molecules
DSSP		for computing secondary structure information that is required by the FNC approach
SWIG	2.0.8	for compiling the *pyFIRST* interface module

Constraint Network Analysis Is the Core Module. The *CNAnalysis* module is the core of the CNA software. The *CNAnalysis* module consists of a single class ConstraintNetworkAnalysis. Upon creating an instance of the type ConstraintNetworkAnalysis, it (I) parses the command line options that specify the analysis type, (II) checks whether the values of the command line arguments conform to the desired data-type, and (III) performs the requested analysis. Depending on the type of analysis, the ConstraintNetworkAnalysis instance creates an instance of the class Dilution if the analysis of a *single network topology* is requested. Otherwise, it creates an instance of the class Fnc or Ensemble, which then creates instances of the class Dilution for each network of the ensemble. The command line options provided by the user are checked for validity by the module *Parameter*; this module also contains default values for the options and internal constants.

***PyFIRST* as an Interface.** We developed the *pyFIRST* interface module to directly access the functionality of the FIRST software (available at http://flexweb.asu.edu) within the Python environment of CNA. The interface module was implemented using the SWIG (Simplified Wrapper and Interface Generator) software tool (http://www.swig.org/).[48] SWIG automatically generates a wrapper code for C/C++ programs that then acts as an interface for other high level programming languages such as Python. The SWIG interface file is written in C++ and contains a single class pyFIRST. The class contains methods that are later on accessible within the Python environment of CNA. Upon instantiating a pyFIRST object, a data structure is generated that represents the constraint network topology of the input structure. Additionally, the pyFIRST object provides methods that are used to (I) read constraint information (covalent bonds, hydrogen bonds, salt bridges, hydrophobic tethers, and stacking interactions) from the network topology, (II) remove constraints from the network with respect to all or a certain type of constraints, and

(III) perform a rigid cluster decomposition. Finally, methods are available that return warnings issued by FIRST when initializing the data structure for the constraint network topology. Note that the *pyFIRST* interface module has been written such that it can be used in any other Python-based application requiring a rigidity analysis by FIRST, thus providing a general Python interface to FIRST.

Structural Information as Input. Single or multiple (in case of a conformational ensemble) input structures for CNA must be in PDB format.[49] Although the validity of the input structure(s) is checked upon creating an instance of the class PDB, we recommend subjecting only complete structures without missing residues or atoms. Hydrogen atoms must be present, too, because otherwise the identification of hydrogen bond and salt bridge constraints cannot be performed. Ligand molecules, if present, are extracted from the input structure and, subsequently, analyzed to determine the bond order by means of Open Babel.[46,47] The last step requires the presence of hydrogen atoms at the ligand. All identified rotatable bonds (single bonds) are then modeled by five bars, whereas nonrotatable bonds (double, triple, amide, and aromatic bonds) are modeled by six bars.[15] Finally, the covalent constraint information for the ligand is merged with the covalent and noncovalent constraint network of the biomacromolecule also generating noncovalent constraints between them. Ions, water, and buffer molecules are handled by FIRST. If an NMR structure is used as input, only the first model is considered. Furthermore, Amber-conform residue names (HIE, HID, HIP, and CYX) are replaced by standard residue names (HIS and CYS) in order to allow the use of PDB structures extracted from molecular dynamics (MD) trajectories created by the Amber software.[50] In the case of a conformational ensemble, a PDB object is instantiated for each conformation. Apart from checking the validity of and preparing the input structure, the PDB class provides several functions that can be used to work with the structure in terms of getting single atom and residue objects, finding neighbor residues within a certain distance cutoff, and writing out structures (including biomacromolecules and ligand molecules) in the PDB format.

Accessing the Network Topology. The *output_network* and *input_network* modules of CNA contain the OutputNetwork and InputNetwork class definitions. Upon instantiating an object, these classes are used to write and read the constraint network topology of a single structure or of each conformation of an ensemble. This is particularly useful for adding user-defined constraints that are not identified automatically, for example, constraints between ions and protein atoms. In the file containing the constraint network topology, each entry of a covalent constraint contains the identifiers of the involved atoms and number of bars of the constraint. For constraints representing hydrophobic or stacking interactions, in addition to the atom identifiers, the distance between the atoms is given plus an indicator whether the constraint occurs within a protein or between protein and ligand. For hydrogen bond and salt bridge constraints, the energy and type of interaction is written instead of the distance and indicator. This file can be modified and used as input for CNA again. In this case, user-defined constraints will overwrite constraint information identified from the input structure(s).

Thermal Unfolding Simulation. The thermal unfolding simulation allows analyzing changes in the network stability upon removing hydrogen bond (including salt bridge) constraints from the network.[4,14,28] To do so, the energy of a hydrogen bond E_{HB} is determined by an empirical energy function.[51] Then, during the thermal unfolding simulation,[4,28] intermediate networks σ are created such that hydrogen bonds with an energy $E_{HB} > E_{cut}(\sigma)$ are removed from the network.[51] This follows the idea that stronger hydrogen bonds will break at higher temperatures than weaker ones. By means of an empirically determined linear function, E_{cut} can be related to a temperature T.[28]

Consequently, the simulation mimics a rise in the temperature by analyzing a range of networks having many hydrogen bonds (equivalent to low temperatures) to having few hydrogen bonds (equivalent to high temperatures). Note that the temperatures should be considered relative values only because the absolute values may depend on the size and architecture of the analyzed protein.[4] Still, the temperatures are very helpful, for example, when it comes to comparing the thermostability of two or more homologous proteins or the stability of a wild-type with its mutant.[4,28,35] An alternative concept grounded in mean-field theory directly connects network rigidity and absolute temperature; while appealing, it requires experimental data for a system-specific parametrization.[20,40] Each of the intermediate networks σ is then subjected to rigidity analysis by FIRST. While the principal idea of the thermal unfolding simulation has been adapted from the FIRST software,[13] the method implemented here allows for additional settings that are not available in the FIRST implementation. These include specifying the energy range and step-size for removing hydrogen bonds. Furthermore, a modified method has been implemented that also considers the temperature dependence of hydrophobic tethers along the thermal unfolding simulation.[35] This approach follows the idea that hydrophobic interactions become stronger with increasing T.[52,53] Accordingly, more hydrophobic tethers are added to the network by linearly increasing the distance cutoff for including hydrophobic tethers $D_{cut}(\sigma)$ from a starting value of 0.25 Å at 300 K to an ending value of 0.40 Å at 420 K. Doing this has been shown to improve thermostability predictions of citrate synthases.[35]

The thermal unfolding simulation is done by the *dilution* module containing the Dilution class. Upon instantiating an object of this class, the object creates new intermediate networks σ and passes the networks through FIRST by instantiating a pyFirst object. Subsequently the module *networkAnalysis* is used to calculate the global and local indices (see section Analyzing the Results from the Rigidity Analysis). Via the global indices, phase transition(s) are identified by an object of the class Transitions. Finally, unfolding nuclei are identified by an object of the class UnfoldingNuclei.

Analyzing the Results from the Rigidity Analysis. The *network_analysis* module comprises in total four classes that process the results from the FIRST rigidity analysis. The main class NetworkAnalysis contains methods to calculate the size and size distribution of rigid clusters and to identify the actual largest rigid cluster as well as the giant percolating cluster of the network. The giant percolating cluster is the largest rigid cluster present at the highest E_{cut} value (i.e., at the lowest temperature) with all constraints in place. During the thermal unfolding simulation, the melting of the giant percolating cluster is monitored, and the largest rigid subcluster of the previous giant percolating cluster becomes the new giant percolating cluster of the present network state σ. Subsequently, the NetworkAnalysis object is passed to three classes for calculating the global and

local indices called GlobalIndices, LocalIndices, and LocalStabilityMaps.

The class GlobalIndices contains all methods that are required to calculate the floppy mode density Φ, the rigidity order parameter P_∞, the cluster configuration entropy H, and the mean rigid cluster size S.[36] Apart from this, the class GlobalIndices also instantiates objects of the classes Transitions and UnfoldingNuclei that are required for the identification of phase transition points and unfolding nuclei of the structure. For identifying phase transition points, two methods have been implemented that make use of the data of the global indices: fitting of a mono/double sigmoid curve and interpolating with a smoothed spline. By default, phase transition points are identified by the double sigmoid curve.[35] However, the user can choose as an option that Akaike's information criterion[54] be used to identify whether a mono or double sigmoid curve gives better fitting results. Finally, if more than two phase transitions are expected or shall be identified, interpolation with the smoothed spline is recommended. Multiple transitions can occur in multimeric proteins. The transition point is then identified for each global index as the point at which the maximal rigidity loss occurs in the structure. Occasionally, a Transitions object does not return a transition point; this occurs if no "sharp" transition can be detected or if multiple transitions with comparable rigidity losses are present.

The class LocalIndices is used to calculate the percolation index p_i and the rigidity index r_i. Both reflect structural stability on a per-residue basis[36] and, thus, can be used to identify the location and distribution of structurally weak or strong parts in biomacromolecules. Finally, the class LocalStabilityMaps is used to calculate the two-dimensional itemization of the rigidity index r_i, the stability map, and a so-called "neighbor stability map", where values of the stability map of residue pairs separated by more than 5 Å are masked. That way, the latter map provides useful information about the stability of neighboring residues only, which can be used for focusing on short-range weak and strong connections within a biomacromolecule.

Writing the Analysis Results. The module *output_results* is used to write results files containing information about global and local indices, phase transition points, and unfolding nuclei. For a phase transition point, the hydrogen bond energy cutoff E_{cut} and the respective temperature are listed. Unfolding nuclei are written out as a text file and PDB file; in the latter, the B-factor column is used to record whether or not a residue is an unfolding nucleus by setting the values to one or zero. If the analysis is performed on an *ensemble of network topologies*, an additional file summarizing the average local indices and standard deviations is written. Similarly, for the phase transition points, mean, median, and standard error are provided in addition. Furthermore, the percentage of network topologies in which a residue is predicted to be an unfolding nucleus is recorded.

Showcase Example: Flexibility Characteristics of HEWL. In a showcase example, we applied the CNA software to a HEWL structure. We show the results for two analysis types, analyzing a single network topology derived from a single input structure (PDB ID: 3LZT) and analyzing an ensemble of network topologies derived from a conformational ensemble. The conformational ensemble was generated by extracting 1500 conformations from a trajectory of 300 ns length obtained by MD simulations starting from an X-ray structure of HEWL (PDB ID: 3LZT). The MD simulation was carried out in explicit solvent at 300 K with the AMBER 11 package of molecular simulation programs.[50] The detailed simulation protocol is described elsewhere (C. Pfleger, H. Gohlke, to be published elsewhere). Water molecules were removed from each conformation before the ensemble was subjected to CNA. Analyzing a single network topology took about 40 s, and the ensemble of 1500 conformations required ~11 h on a single-core workstation computer, which demonstrates the computational efficiency of CNA and FIRST.

Snapshots from the thermal unfolding simulation of the single input structure are depicted in Figure 3. They show the

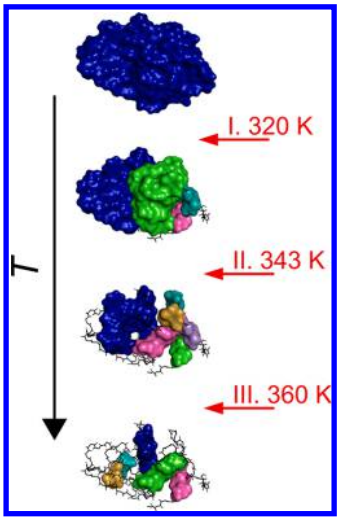


Figure 3. Rigid cluster decompositions along the thermal unfolding simulation of the showcase example HEWL. Rigid clusters are shown as uniformly colored bodies connected by flexible hinge regions (black). The roman numbers relate to three major steps of rigidity loss.

loss of rigidity in terms of the decay of rigid clusters with increasing temperature. The first transition relates to the beginning of the collapse of the giant rigid cluster, which occurs in the interface region of the α- and β-domains. At this state, the network is dominated by two large rigid components. During the next transition, the rigid cluster covering the α-domain collapses, and the helical elements remain as single rigid clusters. Finally, during the last transition, the rigid cluster covering the β-domain collapses, and nearly the whole system becomes flexible. The results from the thermal unfolding simulation agree, in reverse order, with the "fast track" folding pathway described in refs 55 and 56. Here, both domains of HEWL fold concurrently but with a slight preference to initially form native contacts in the β-domain.[57] Alternatively, a "slow track" folding reaction of HEWL has been described,[56,58,59] in which the majority of the protein molecules populate an intermediate state with persistent structures in only the α-domain.[57] Still, parts of the α-domain need to unfold again to enable the subsequent folding of the β-domain.

As an example for a global index, the cluster configuration entropy H is shown, which monitors the loss of network stability during the thermal unfolding simulation. In the analysis of the single network topology (Figure 4a), an early phase transition at 319 K indicates the beginning decay of structural stability, with most of the network still being captured in rigid clusters. The dominant phase transition at 343 K then refers to the point at which the network loses its ability to carry stress

dx.doi.org/10.1021/ci400044m | J. Chem. Inf. Model. 2013, 53, 1007−1015

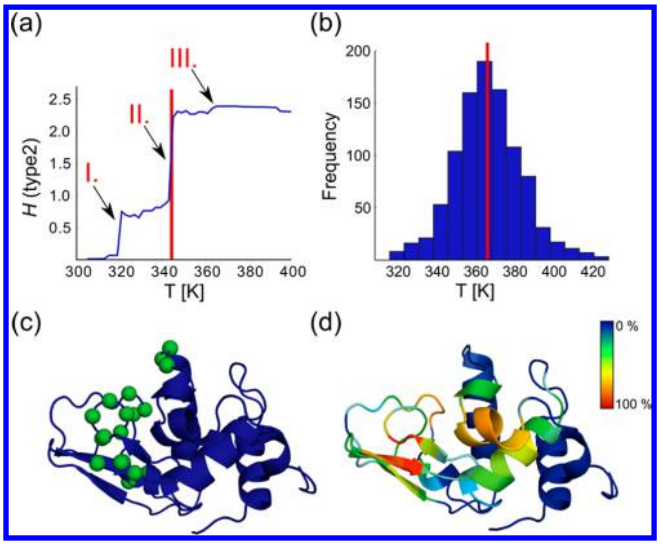


Figure 4. (a) Cluster configuration entropy H (type 2) derived from the single network topology. The entropy is plotted as a function of the temperature, and the roman numbers correspond to the three major steps depicted in Figure 3. The phase transition automatically identified by CNA is marked by the red vertical line. (b) Frequency distribution of phase transitions identified from analyzing the ensemble of network topologies. The median is marked with a red vertical line. (c) Weak spot detection for the single network topology. Green spheres highlight the identified weak spot residues in the HEWL structure. (d) Weak spot detection over the ensemble of network topologies. For depicting the probability of being a weak spot, each residue is colored according to a color scale ranging from blue (low probability) to red (high probability).

and, hence, corresponds to the folded−unfolded transition. The last transition indicates the loss of the remaining rigid components. In the case of analyzing the ensemble of network topologies, the frequency distribution of the identified phase transition points is shown (Figure 4b). From this, a median transition temperature of 358 K is revealed, which is 15 K higher than the dominant phase transition point identified from analyzing a single network topology. Note that, in general, phase transitions identified using a single input structure can be different from ensemble results, as shown in a previous study on citrate synthase.[35] We thus recommend performing CNA analyses on ensembles of network topologies, in particular, when quantitative results are desired. At the transition point, unfolding nuclei are identified (Figure 4c). Almost all unfolding nuclei are located in the β-domain of HEWL, which disintegrates at the dominant phase transition (Figures 3 and 4c). Furthermore, for the ensemble of network topologies, the probability of a residue being found as an unfolding nucleus over the entire ensemble is provided (Figure 4d). The higher this probability the more likely will it be that rigidifying this residue will improve protein stability. The ensemble results are more detailed than the ones from the single structure in that now unfolding nuclei are not only located in the β-domain but also in helix B, which agrees with the view that this helix plays a crucial role in stabilizing the tertiary structure of HEWL.[60]

As for local indices, we exemplary show the rigidity index r_i, which characterizes the stability of the HEWL structure down to the bond level (Figure 5a, b). As such, r_i monitors the point when a residue segregates from a rigid cluster along the thermal unfolding simulation: the lower r_i the longer is a residue part of a rigid cluster. Secondary structure elements are generally found to be more stable than loop regions. Furthermore, averaging r_i values over the ensemble of network topologies leads to a smoother r_i curve and to the spike located at residue 78 becoming less pronounced than in the case of analyzing the single network topology. The spike reveals a region that is highly stabilized by hydrophobic interactions; these regions only melt at a late stage of the thermal unfolding simulations. Notably, the stable regions identified for residues 53 and 62−65 are in very good agreement with those identified by high protection factors in H/D experiments for the native and denatured states of HEWL.[61] During the catalytic cycle, HEWL undergoes a reorientation of the α- and β-domains due to a bending movement around a central hinge region.[62] Along these lines, the identified flexible hinge regions (Figure 5a, b) are in agreement with those suggested by McCammon et al.[62] and coincide with results obtained from Gaussian network models and MD simulations.[60,63] Such a decomposition into rigid clusters and flexible regions is used as a first step in a normal mode-based geometric simulation approach (NMSim) working on a coarse-grained protein representation.[64] With this, stereochemically and energetically favorable conformations of HEWL were generated previously.[64]

As yet another local index, stability maps rc_{ij} are two-dimensional itemizations of the r_i and report when a "rigid contact" between two residues of the network vanishes during the thermal unfolding simulation. The upper triangles of Figure 5c and d show the stability maps for the single network topology and the ensemble of network topologies, respectively. Again, blocks of stable contacts are pronounced for secondary structures elements. In contrast, very weak contacts are identified for residues 81−87 that partially form a 3_{10} helix. This is in agreement with results from NMR experiments that reveal a disordered structure of this region.[65] The lower triangles of Figure 5c and d show a modification of the stability map that highlights solely those residue pairs with a "rigid contact" where the residues are within a distance of 5 Å. This map is referred to as "neighbor stability map". Accordingly, a rigid contact in such a map that melts early in the thermal unfolding simulation is a prominent target for rigidification and, hence, for improving protein stability.

■ CONCLUSIONS

In recent years, there has been encouraging progress in characterizing the flexibility and rigidity of biomacromolecules down to the residue level by graph theoretical approaches. However, for deriving maximal advantage from information on biomacromolecular flexibility and rigidity, results from rigidity analyses must be linked to biologically relevant characteristics of a structure, such as (thermo-)stability and function. This provided the incentive for us to develop the CNA software package presented here. CNA functions as a front- and backend to the FIRST software and allows setting up a variety of constraint network representations, processing the results obtained from FIRST, and calculating global and local indices for quantifying biomacromolecular stability.

Thus, while CNA relies on FIRST as a core engine, it goes beyond the mere identification of flexible and rigid regions in a biomacromolecular structure. Major advancements in that respect include (I) a refined modeling of thermal unfolding simulations that considers the temperature-dependence of hydrophobic tethers, (II) the ability to perform rigidity analyses on ensembles of network topologies, either generated from structural ensembles provided as input or by using the concept

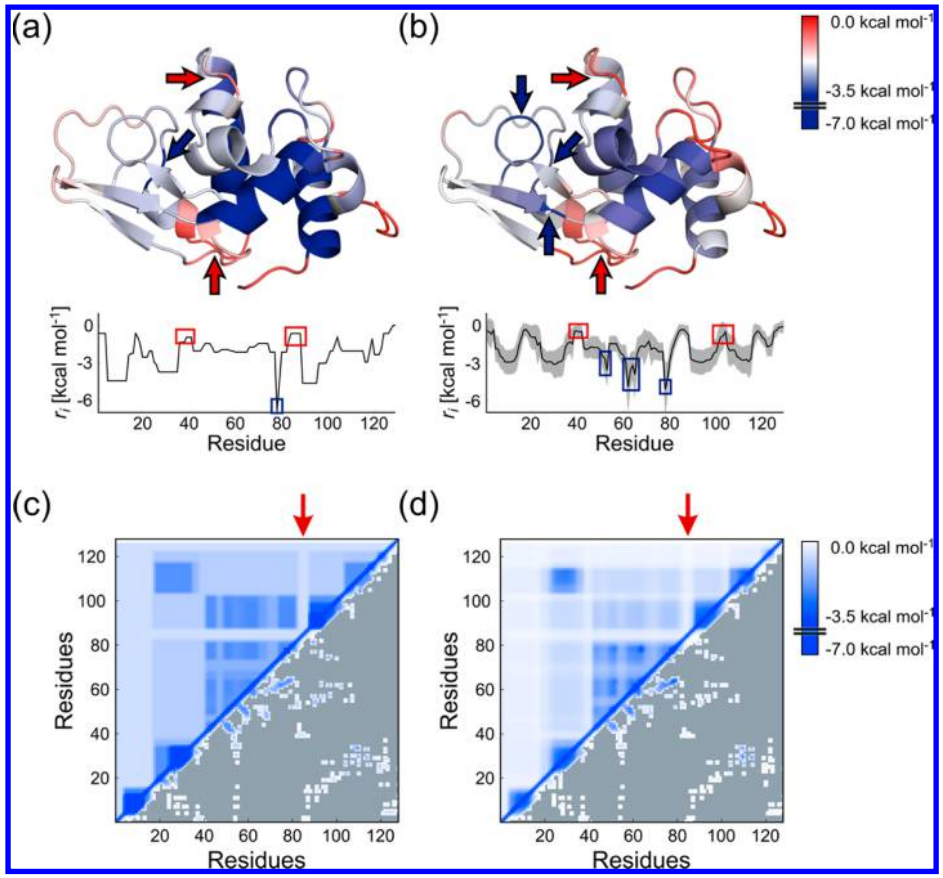


Figure 5. (a) Rigidity index r_i determined by analyzing the single network topology and (b) ensemble of network topologies plotted against a residue identifier and color coded onto the structure (range of color code: red (flexible) to blue (rigid)). In addition, the plot in (b) shows the standard deviation as a gray area. Blue rectangles and blue arrows in panels (a) and (b) highlight structurally stable regions for which high protection factors have been determined by H/D experiments. Red rectangles and red arrows in panels (a) and (b) highlight structurally flexible regions that are associated with hinge regions of HEWL. Stability maps (upper triangle) and neighbor stability maps (lower triangle) determined by analyzing the single network topology (c) and the ensemble of network topologies (d). The color depicts how stably two residues are connected and ranges from white (low stability) to blue (high stability). Red arrows highlight regions that reveal a disordered structure in NMR experiments. Gray areas in the neighbor stability map are displayed when residues are more than 5 Å away from each other.

of fuzzy noncovalent constraints, and (III) computing a set of global and local indices for characterizing biomacromolecular flexibility and rigidity, three of which have been introduced only recently by us.[36] The advancements allow (I) modeling in a more detailed manner the thermal unfolding of biomacromolecules, (II) obtaining more robust results from rigidity analyses due to a reduced sensitivity to the structural input, and (III) extending the application domain of rigidity analyses in that phase transition points ("melting points") and unfolding nuclei ("structural weak spots") are determined automatically. Such advancements are important for data-driven protein engineering, for example, for identifying structural parts that influence protein thermostability.[28] Furthermore, CNA robustly handles small-molecule ligands in general. This is important when it comes to estimating the influence of ligands on biomacromolecular stability, for example, for probing signal transmission across a protein structure for understanding and predicting "dynamic allostery"[66] and in assessing (changes in) flexibility characteristics of binding sites and interface regions.[67] How CNA can be applied in that respect has been demonstrated in a showcase example on HEWL.

CNA maintains the efficiency of FIRST. This has been achieved by linking CNA and FIRST via the *pyFIRST* interface module, minimizing the I/O overhead. The analysis of a single protein structure by CNA usually takes only a few seconds for systems of several hundred residues on a single core. The runtime for analyses of ensembles of network topologies, which is in the order of hours currently, could be further reduced given that processing individual members of such an ensemble is trivially parallelizable. Finally, the hierarchical design of the software makes CNA highly adaptable and extensible, for example, by adding new index definitions.

Overall, we believe that these unique features make CNA an interesting tool for linking biomacromolecular structure, flexibility, (thermo-)stability, and function.

AUTHOR INFORMATION

Corresponding Author

* Phone: (+49) 211-81-13662. Fax: (+49) 211-81-13847. E-mail: gohlke@uni-duesseldorf.de.

Present Address

[†]Elsevier Information Systems GmbH, Frankfurt am Main, Germany.

Author Contributions

[‡]These authors contributed equally to this work.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We are grateful to the Ministry of Innovation, Science, and Research of North Rhine-Westphalia and Heinrich-Heine-University Düsseldorf for a scholarship to PCR within the CLIB-Graduate Cluster Industrial Biotechnology. We are grateful to Daniel Mulnaes (Heinrich-Heine-University, Düsseldorf) for proofreading the manuscript. CNA is available from http://cpclab.uni-duesseldorf.de/software for nonprofit organizations.

■ ABBREVIATIONS

CNA, Constraint Network Analysis; FIRST, Floppy Inclusions and Rigid Substructure Topography; HEWL, Hen Egg White Lysozyme; FNC, Fuzzy Noncovalent Constraints; PDB, Protein Data Bank; DSSP, Define Secondary Structure of Proteins

■ REFERENCES

(1) Ahmed, A.; Kazemi, S.; Gohlke, H. Protein flexibility and mobility in structure-based drug design. *Front. Drug Des. Discovery* **2007**, *3*, 455−476.
(2) Heal, J. W.; Jimenez-Roldan, J. E.; Wells, S. A.; Freedman, R. B.; Romer, R. A. Inhibition of HIV-1 protease: The rigidity perspective. *Bioinformatics* **2012**, *28*, 350−357.
(3) Jagodzinski, F.; Hardy, J.; Streinu, I. Using rigidity analysis to probe mutation-induced structural changes in proteins. *J. Bioinf. Comput. Biol.* **2012**, 10.
(4) Radestock, S.; Gohlke, H. Protein rigidity and thermophilic adaptation. *Proteins* **2011**, *79*, 1089−1108.
(5) Tan, H. P.; Rader, A. J. Identification of putative, stable binding regions through flexibility analysis of HIV-1 gp120. *Proteins* **2009**, *74*, 881−894.
(6) Halle, B. Flexibility and packing in proteins. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99*, 1274−1279.
(7) Dokholyan, N. V.; Li, L.; Ding, F.; Shakhnovich, E. I. Topological determinants of protein folding. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99*, 8637−8641.
(8) Vendruscolo, M.; Dokholyan, N. V.; Paci, E.; Karplus, M. Small-world view of the amino acids that play a key role in protein folding. *Phys. Rev. E* **2002**, *65*, 1−4.
(9) Böde, C.; Kovács, I. A.; Szalay, M. S.; Palotai, R.; Korcsmáros, T.; Csermely, P. Network analysis of protein dynamics. *FEBS Lett.* **2007**, *581*, 2776−2782.
(10) Greene, L. H.; Higman, V. A. Uncovering network systems within protein structures. *J. Mol. Biol.* **2003**, *334*, 781−791.
(11) Heringa, J.; Argos, P. Side-chain clusters in protein structures and their role in protein folding. *J. Mol. Biol.* **1991**, *220*, 151−171.
(12) Heringa, J.; Argos, P.; Egmond, M. R.; Devlieg, J. Increasing thermal stability of subtilisin from mutations suggested by strongly interacting side-chain clusters. *Protein Eng.* **1995**, *8*, 21−30.
(13) Jacobs, D. J.; Rader, A. J.; Kuhn, L. A.; Thorpe, M. F. Protein flexibility predictions using graph theory. *Proteins* **2001**, *44*, 150−165.
(14) Rader, A. J.; Hespenheide, B. M.; Kuhn, L. A.; Thorpe, M. F. Protein unfolding: Rigidity lost. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99*, 3540−3545.
(15) Whiteley, W. Counting out to the flexibility of molecules. *Phys. Biol.* **2005**, *2*, S116−S126.
(16) Jacobs, D. J.; Thorpe, M. F. Generic rigidity percolation: The pebble game. *Phys. Rev. Lett.* **1995**, *75*, 4051−4054.
(17) Jacobs, D. J.; Hendrickson, B. An algorithm for two-dimensional rigidity percolation: The pebble game. *J. Comput. Phys.* **1997**, *137*, 346−365.
(18) Katoh, N.; Tanigawa, S. A proof of the molecular conjecture. *Discrete Comput. Geom.* **2011**, *45*, 647−700.
(19) Hespenheide, B. M.; Rader, A. J.; Thorpe, M. F.; Kuhn, L. A. Identifying protein folding cores from the evolution of flexible regions during unfolding. *J. Mol. Graphics Modell.* **2002**, *21*, 195−207.
(20) Jacobs, D. J.; Dallakyan, S.; Wood, G. G.; Heckathorne, A. Network rigidity at finite temperature: Relationships between thermodynamic stability, the nonadditivity of entropy, and cooperativity in molecular systems. *Phys. Rev. E* **2003**, 68.
(21) Gohlke, H.; Kuhn, L. A.; Case, D. A. Change in protein flexibility upon complex formation: Analysis of Ras-Raf using molecular dynamics and a molecular framework approach. *Proteins* **2004**, *56*, 322−337.
(22) Rader, A. J.; Anderson, G.; Isin, B.; Khorana, H. G.; Bahar, I.; Klein-Seetharaman, J. Identification of core amino acids stabilizing rhodopsin. *Proc. Natl. Acad. Sci. U.S.A.* **2004**, *101*, 7246−7251.
(23) Rader, A. J.; Bahar, I. Folding core predictions from network models of proteins. *Polymer* **2004**, *45*, 659−668.
(24) Mamonova, T.; Hespenheide, B.; Straub, R.; Thorpe, M. F.; Kurnikova, M. Protein flexibility using constraints from molecular dynamics simulations. *Phys. Biol.* **2005**, *2*, S137−S147.
(25) Wells, S.; Menor, S.; Hespenheide, B. M.; Thorpe, M. F. Constrained geometric simulation of diffusive motion in proteins. *Phys. Biol.* **2005**, *2*, S127−S136.
(26) Livesay, D. R.; Jacobs, D. J. Conserved quantitative stability/flexibility relationships (QSFR) in an orthologous RNase H pair. *Proteins* **2006**, *62*, 130−143.
(27) Ahmed, A.; Gohlke, H. Multiscale modeling of macromolecular conformational changes combining concepts from rigidity and elastic network theory. *Proteins* **2006**, *63*, 1038−1051.
(28) Radestock, S.; Gohlke, H. Exploiting the link between protein rigidity and thermostability for data-driven protein engineering. *Eng. Life Sci.* **2008**, *8*, 507−522.
(29) Fulle, S.; Gohlke, H. Analyzing the flexibility of RNA structures by constraint counting. *Biophys. J.* **2008**, *94*, 4202−4219.
(30) Fulle, S.; Gohlke, H. Constraint counting on RNA structures: Linking flexibility and function. *Methods* **2009**, *49*, 181−188.
(31) Fulle, S.; Gohlke, H. Statics of the ribosomal exit tunnel: Implications for cotranslational peptide folding, elongation regulation, and antibiotics binding. *J. Mol. Biol.* **2009**, *387*, 502−517.
(32) Fulle, S.; Christ, N. A.; Kestner, E.; Gohlke, H. HIV-1 TAR RNA spontaneously undergoes relevant apo-to-holo conformational transitions in molecular dynamics and constrained geometrical simulations. *J. Chem. Inf. Model.* **2010**, *50*, 1489−1501.
(33) Mottonen, J. M.; Jacobs, D. J.; Livesay, D. R. Allosteric response is both conserved and variable across three CheY orthologs. *Biophys. J.* **2010**, *99*, 2245−2254.
(34) Rader, A. J. Thermostability in rubredoxin and its relationship to mechanical rigidity. *Phys. Biol.* **2010**, *7*, 016002.
(35) Rathi, P. C.; Radestock, S.; Gohlke, H. Thermostabilizing mutations preferentially occur at structural weak spots with a high mutation ratio. *J. Biotechnol.* **2012**, *159*, 135−144.
(36) Pfleger, C.; Radestock, S.; Schmidt, E.; Gohlke, H. Global and local indices for characterizing biomolecular flexibility and rigidity. *J. Comput. Chem.* **2013**, *34*, 220−233.
(37) Wells, S. A.; Jimenez-Roldan, J. E.; Romer, R. A. Comparative analysis of rigidity across protein families. *Phys. Biol.* **2009**, *6*.
(38) Zaccai, G. Biochemistry - How soft is a protein? A protein dynamics force constant measured by neutron scattering. *Science* **2000**, *288*, 1604−1607.
(39) Jacobs, D. J.; Dallakyan, S. Elucidating protein thermodynamics from the three-dimensional structure of the native state using network rigidity. *Biophys. J* . **2005**, *88*, 903−915.
(40) Gonzalez, L. C.; Wang, H.; Livesay, D. R.; Jacobs, D. J. Calculating ensemble averaged descriptions of protein rigidity without sampling. *PLoS One* **2012**, 7.
(41) Cock, P. J.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; de Hoon, M. J. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **2009**, *25*, 1422−1423.
(42) Joosten, R. P.; Beek, T. A. H. T.; Krieger, E.; Hekkelman, M. L.; Hooft, R. W. W.; Schneider, R.; Sander, C.; Vriend, G. A series of PDB

related databases for everyday needs. *Nucleic Acids Res.* **2011**, *39*, D411−D419.
(43) Kabsch, W.; Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **1983**, *22*, 2577−2637.
(44) Ascher, D.; Dubois, P. F.; Hinsen, K.; Hugunin, J.; Oliphant, T. *Numerical Python*, 2001.
(45) Jones, E.; Oliphant, T.; Peterson, P. *SciPy: Open Source Scientific tools for Python*, 2001
(46) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J. Cheminform.* **2011**, 3.
(47) O'Boyle, N. M.; Morley, C.; Hutchison, G. R. Pybel: A Python wrapper for the OpenBabel cheminformatics toolkit. *Chem. Cent. J.* **2008**, 2.
(48) Beazley, D. M. Automated scientific software scripting with SWIG. *Future Gener. Comput. Syst.* **2003**, *19*, 599−609.
(49) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235−242.
(50) Case, D.A.; T. A. D., Cheatham, T.E.; , III, Simmerling, C.L.; Wang, J.; Duke, R.E.; Luo, R.; Walker, R.C.; Zhang, W.; Merz, K.M.; Roberts, B.; Wang, B.; Hayik, S.; Roitberg, A.; Seabra, G.; I. Kolossváry, Wong, K.F.; Paesani, F.; Vanicek, J.; Liu, J.; Wu, X.; Brozell, S.R.; Steinbrecher, T.; Gohlke, H.; Cai, Q.; Ye, X.; Wang, J.; Hsieh, M.-J.; Cui, G.; Roe, D.R.; Mathews, D.H.; Seetin, M.G.; Sagui, C.; Babin, V.; Luchko, T.; Gusarov, S.; Kovalenko, A.; Kollman, P.A. *AMBER 11*; University of California: San Francisco, 2010.
(51) Dahiyat, B. I.; Gordon, D. B.; Mayo, S. L. Automated design of the surface positions of protein helices. *Protein Sci.* **1997**, *6*, 1333−1337.
(52) Privalov, P. L.; Gill, S. J. Stability of protein−structure and hydrophobic interaction. *Adv. Protein Chem.* **1988**, *39*, 191−234.
(53) Schellman, J. A. Temperature, stability, and the hydrophobic interaction. *Biophys. J.* **1997**, *73*, 2960−2964.
(54) Burnham, K. P.; Anderson, D. R. *Model Selection and Multimodel Inference: A Practical Information−Theoretic Approach*, 2. ed.; Springer: New York, 2002; pp XXVI, 488 S.
(55) Radford, S. E.; Dobson, C. M.; Evans, P. A. The folding of hen lysozyme involves partially structured intermediates and multiple pathways. *Nature* **1992**, *358*, 302−307.
(56) Matagne, A.; Radford, S. E.; Dobson, C. M. Fast and slow tracks in lysozyme folding: Insight into the role of domains in the folding process. *J. Mol. Biol.* **1997**, *267*, 1068−1074.
(57) Dinner, A. R.; Sali, A.; Smith, L. J.; Dobson, C. M.; Karplus, M. Understanding protein folding via free-energy surfaces from theory and experiment. *Trends Biochem. Sci.* **2000**, *25*, 331−339.
(58) Kiefhaber, T. Kinetic traps in lysozyme folding. *Proc. Natl. Acad. Sci. U.S.A.* **1995**, *92*, 9029−9033.
(59) Wildegger, G.; Kiefhaber, T. Three-state model for lysozyme folding: Triangular folding mechanism with an energetically trapped intermediate. *J. Mol. Biol.* **1997**, *270*, 294−304.
(60) Haliloglu, T.; Bahar, I. Structure-based analysis of protein dynamics: Comparison of theoretical results for hen lysozyme with X-ray diffraction and NMR relaxation data. *Proteins* **1999**, *37*, 654−667.
(61) Radford, S. E.; Buck, M.; Topping, K. D.; Dobson, C. M.; Evans, P. A. Hydrogen-exchange in native and denatured states of hen egg-white lysozyme. *Proteins* **1992**, *14*, 237−248.
(62) McCammon, J. A.; Gelin, B. R.; Karplus, M.; Wolynes, P. G. The hinge-bending mode in lysozyme. *Nature* **1976**, *262*, 325−6.
(63) Kohn, J. E.; Afonine, P. V.; Ruscio, J. Z.; Adams, P. D.; Head-Gordon, T. Evidence of functional protein dynamics from X-ray crystallographic ensembles. *PLoS Comput. Biol.* **2010**, 6.
(64) Ahmed, A.; Rippmann, F.; Barnickel, G.; Gohke, H. A normal mode-based geometric simulation approach for exploring biologically relevant conformational transitions in proteins. *J. Chem. Inf. Model.* **2011**, *51*, 1604−1622.
(65) Smith, L. J.; Sutcliffe, M. J.; Redfield, C.; Dobson, C. M. Structure of hen lysozyme in solution. *J. Mol. Biol.* **1993**, *229*, 930−944.
(66) Tzeng, S. R.; Kalodimos, C. G. Dynamic activation of an allosteric regulatory protein. *Nature* **2009**, *462*, 368−U139.
(67) Metz, A.; Pfleger, C.; Kopitz, H.; Pfeiffer-Marek, S.; Baringhaus, K. H.; Gohlke, H. Hot spots and transient pockets: Predicting the determinants of small-molecule binding to a protein−protein interface. *J. Chem. Inf. Model.* **2011**, *52*, 120−133.
(68) Krüger, D. M.; Rathi, P. C.; Pfleger, C.; Gohlke, H. CNA Web server: Rigidity theory-based thermal unfolding simulations of proteins for linking structure, (thermo)stability, and function. *Nucleic Acids Res.* **2013**, DOI: 10.1093/nar/gkt292.

NOTE ADDED IN PROOF

A CNA Web server is available at http://cpclab.uni-duesseldorf.de/cna/.[68]

NOTE ADDED AFTER ASAP PUBLICATION

This paper was published ASAP on April 8, 2013, with an error in reference 68. The corrected version was published to the Web on April 9, 2013.