JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

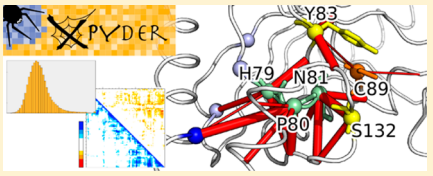
xPyder: A PyMOL Plugin To Analyze Coupled Residues and Their Networks in Protein Structures.

Marco Pasi,[†,‡,§] Matteo Tiberti,[†,§] Alberto Arrigoni,[†] and Elena Papaleo*[,†,∥]

[†]Department of Biotechnology and Biosciences, University of Milano-Bicocca, P.zza della Scienza 2, 20126, Milan, Italy

[‡]Université Lyon 1, CNRS, UMR 5086, Bases Moléculaires et Structurales des Systèmes Infectieux, IBCP FR3302, 7 passage du Vercors, Lyon, F-69367, France

ABSTRACT: A versatile method to directly identify and analyze short- or long-range coupled or communicating residues in a protein conformational ensemble is of extreme relevance to achieve a complete understanding of protein dynamics and structural communication routes. Here, we present xPyder, an interface between one of the most employed molecular graphics systems, PyMOL, and the analysis of dynamical cross-correlation matrices (DCCM). The approach can also be extended, in principle, to matrices including other indexes of communication propensity or intensity between protein residues, as well as the persistence of intra- or intermolecular interactions, such as those underlying protein dynamics. The xPyder plugin for PyMOL 1.4 and 1.5 is offered as Open Source software via the GPL v2 license, and it can be found, along with the installation package, the user guide, and examples, at http://linux.btbs.unimib.it/xpyder/.



1. INTRODUCTION

It is presently well established that in the study of biomolecules it is necessary not only to consider the structure−function paradigm but also the triad structure−function-dynamics.[1−4] In fact, looking at protein dynamics on different time scales, and differentiating between low and high frequency motions, low frequency motions emerge as the most relevant for protein function.[5−7]

They are extremely conserved at the family and superfamily level[5,8−14] and less perturbed by mutations.[15−17] Nevertheless, differences can be traced in protein dynamics and in the paths of structural communication even in protein sharing the same overall 3D architecture. They are promoted by differences in the underlying network of intra- or intermolecular interactions or communicating residues, and they can be related to adaptations to different environmental conditions, differences in biological function, allosteric effects induced by other interaction partners, post-translational modification, or mutations.[10,18−25]

In this context, molecular dynamics (MD) simulations turned out to be a successful technique to study function−structure−dynamics relationships,[1,26−28] thanks to the recent improvements in force field accuracy.[29,30] In fact, MD simulations provide a description of protein dynamics on different time scales in atomistic details and efficiently contribute to rationalize experimental data, in particular if simulations and experiments work in a continuous cross-talk.

Several methods to estimate protein flexibility are available for MD simulations, and they provide a general description of protein dynamics, as root-mean-square fluctuations, B-factor, or generalized order parameters. Other methods have to be evoked to define the atomistic details of protein dynamics and the underlying networks of interactions and communications.[31−37] These methods have found widespread application in recent years, and their coupling to protein structure networks (PSN)[11,21,33,38,39] or coevolution metrics[12,40−43] has led to particularly interesting developments. In fact, they are crucial tools to disclose long-range communication and allosteric effects, which are known to be an intrinsic property of any protein.[44−47]

A notable example to analyses of communications between residues during dynamics is the calculation of dynamical cross-correlation matrices (DCCMs, $C(i,j)$) from the covariance matrix of atomic fluctuations, allowing the detection of both positively correlated, negatively correlated, or uncoupled motions.[48] The method computes the DCCM from the carbon-α (Cα) covariance of atomic positional fluctuations. A $C(i,j) < 0$ corresponds to atoms whose motions along a given spatial coordinate is opposite (anticorrelated motion), while atoms moving along the same direction tend to have a $C(i,j) > 0$ (correlated motion). Several programs presently implement tools to calculate DCCM and also postprocessing analyses of these matrices, but a tool dedicated to the visual inspection and analysis of this data directly on the three-dimensional (3D) structure is still missing. The relevance of studying protein dynamics including concepts from the network theory is even more emphasized by the recent effort in the context of the DynaSIN project of integrating 3D structural data on dynamic conformational changes in the networks of molecular protein−protein interactions.[49] Very recently, the representation of residues interactions according to network theory has been also applied to the visualization of intramolecular interactions in protein structures.[50−54] Moreover, a Web server to calculate and plot results from PSN analysis on a single 3D structure is available.[55]

In principle, any data that can be quantitatively represented as a list of pairwise relationships between residues is eligible to be

Received: May 1, 2012
Published: June 21, 2012

treated in the well-known framework of graph theory as a network of interconnected nodes. The use of the graph formalism permits exploiting the vast amount of data available in structural biology to extract emerging properties from a high number of single relations. For instance, the popular PSN method defines a numeric index representative of the non-bonding interactions for each possible residue pair in an individual conformation or in a conformational ensemble, thus building a PSN graph.[33] Graph analysis was exploited to identify highly connected nodes (hubs), which were often shown to be residues with crucial structural or functional roles and well conserved within protein families and superfamilies.[11,33,39] Moreover, the employment of algorithms for path search allows identifying important communication routes in the protein structure.[33] Similar strategies can be employed on differently defined weighted graphs, as long as the one-value-per-residue-pair rule holds. For example, one could employ a similar treatment on binary hydrophobic or electrostatic interactions calculated from an experimental or MD ensemble to isolate clusters of interacting residues on the protein structure, as recently applied to several cases of study.[21,22,56,57]

In the last 10 years, the PyMOL molecular graphics system (https://www.pymol.org) has evolved its capabilities to become a platform that can be integrated with several programs to provide different and versatile analysis and visualization tools for biomolecular structures. PyMOL is suitable to be extended and customized without recompiling the source code thanks to its multilayer architecture and the use of the object-oriented scripting language Python.[58] Extensions can either take advantage of the wizard interface or employ a system for plugin interfaces, which is more versatile and most commonly used. In fact, several plugins and programs have been developed that carry out different analyses using the PyMOL interface,[59−67] as well as interfaces to integrate PyMOL to docking or virtual screening programs[68−70] or tools to create animations and movies.[71] Recently PyMOL and the popular ab initio modeling program Rosetta have been also successfully integrated.[72]

In this contribution, we present a plugin for PyMOL, xPyder, which enables extensible analyses and a customizable 3D representation of cross-correlations between residues in dynamics and, in general, of pairs of communicating or interacting residues defined by a variety of different methods. xPyder can therefore provide a valuable instrument for the investigation of biomolecular structure and dynamics.

2. METHODS

2.1. Calculation of DCCM and Other Correlation Metrics. Dynamic cross-correlation matrices (DCCM)[48] are generally exploited in order to detect correlated motions in the protein structures. The method generally makes use of the Cα covariance of atomic positional fluctuations, so that the dynamic cross-correlation matrices (DCCM) $C(i,j)$ are computed according to the following expression

$$C(i,j) = \frac{c(i,j)}{c(i,i)^{1/2} c(j,j)^{1/2}}$$

where $c(i,j)$ is the covariance matrix element of protein fluctuation between residues i and j.

The $C(i,j)$ values reported in the matrix are normalized so that they range from −1 to +1. A $C(i,j) < 0$ corresponds to atoms whose motions along a given spatial coordinate are opposite (anticorrelated motion), while atoms moving along the same direction tend to have a $C(i,j) > 0$ (correlated motion). The resulting $n \times n$ cross-correlation matrix is symmetrical along the diagonal. In the analysis of correlated motions, cutoffs are generally employed to filter out the pairs of residues characterized by almost uncoupled motions. Besides the Pearson correlation coefficients, other metrics can be adopted to define correlated motions, and similar formulations can be also employed. In fact, a suitable alternative is to employ a generalized correlation measure on the basis of mutual information (MI), as well as a linearized generalized correlation coefficient that can be derived within the MI framework, allowing for separate linear and nonlinear contributions by Linear Mutual Information (LMI) as described in ref 73.

2.2. Chained Correlations. Chained correlation are derived from postprocessing of the input matrix starting from a root residue or a selection of residues. Chained correlations can help in the identification of long-range communication through the calculation of intermediate correlations.[74]

The calculation of chains of correlated residues is performed on a simple weighted graph defined by considering the filtered loaded input matrix as the graph adjacency matrix. The residues are used as graph vertices and the matrix values as edge weights. No edge is created if the corresponding matrix value is zero or less, thus considering positive weights only. The graph is calculated on the fly each time the user decides to perform the chained analysis.

An algorithm similar to that developed in FlexServ[74] is employed for the calculation of chained correlations. In particular, the algorithm to calculate chained correlations employs a variation of the breadth-first search algorithm.[75] According to this method, a starting (root) residue, width (w), and depth (d) parameters are selected by the user. In particular, w defines the number of residues directly correlated to the root residue (neighbors) that have to be included in the search for chained correlations. The calculation employs a uninformed search procedure: the algorithm starts from the root node and identifies its w neighbors that are connected to the root node by the highest-weight edges (i.e., the w residues featuring the highest correlation values with the root residue). Each of the selected root neighbors is then considered as the new root at the next iteration, and each of their w neighbors is expanded according to the same procedure, ensuring that no node is visited more than once. The search for chained correlations is performed until the selected d value is reached.

It can be useful to previously filter the input matrix with a specific cutoff on matrix values, so that the less relevant correlations are not included in the analysis, and thus, more robust and significant paths can be highlighted. Moreover, it can be informative to perform a first search from a root residue (or selection) using specific w and d values and then to iterate the search for increasing d values until no more residues could be connected to the ones identified at the previous step.

In xPyder, an option is also available to filter the output map (subgraph) of chained correlations by removing the terminal nodes identified in the search, which are not further connected nodes within the subgraph. Therefore, as final output only, the connections for which the further nodes would communicate to a successive node are retained. Another option can be set by the user to exclude in the identification of chained correlated residues the vertices that are connected to the parent node of the current root at the previous iteration. Finally, an option is available to plot among the paths identified from the root residue

Figure 1. xPyder's user interface. Left panel: Plot settings mask of xPyder, which shows the histogram of the loaded matrix and the plot settings for a DCCM; active filters are shown green in the panel on the right. Right panel: Overview of the configurations for the default filtering plugins.

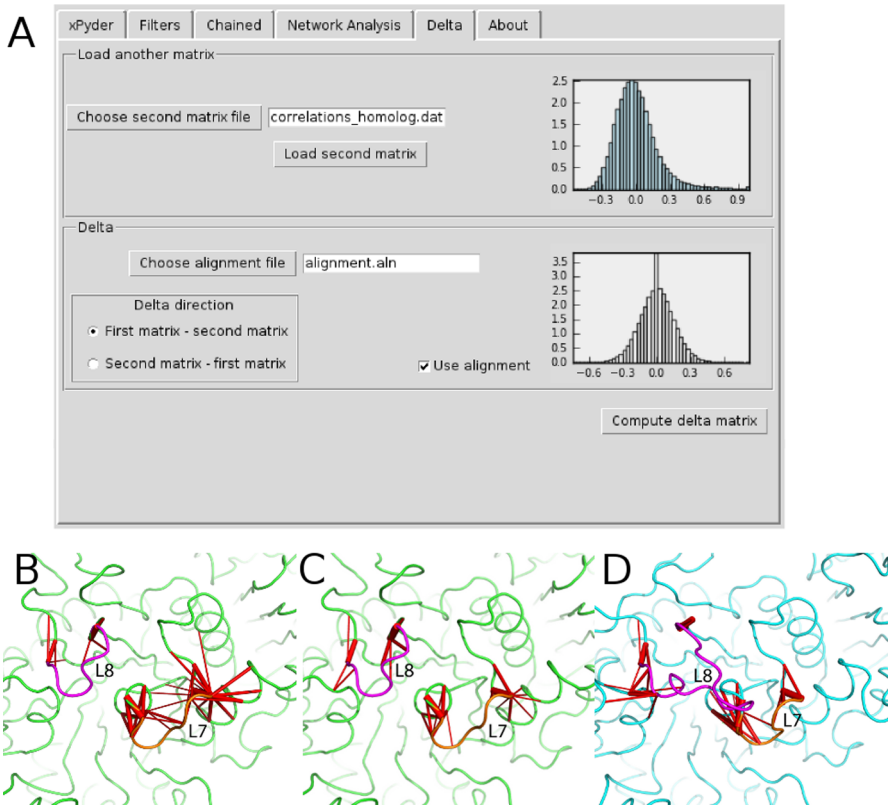


Figure 2. Comparison between dynamical correlations. (A) Delta analysis mask of xPyder, showing the histograms of the second loaded matrix and of the delta matrix. (B–D) Example of comparison between cross-correlated motions in a cold-adapted α-amylase (B, pdb entry 1AQH), its mesophilic-like mutant (C), and the mesophilic homolog (D, pdb entry 1PIF) with particular attention to the loop 7 (L7, orange) and 8 (L8, purple) around the active site.[22] Cross correlations are plotted as cylinders of thickness proportional to the correlation value.

to each terminal node only the one featuring the highest cumulative weight.

2.3. Graph Analysis. xPyder allows carrying out several different analyses on the matrix-encoded graph. First, a simple weighted graph is created, as previously detailed for the chained cross-correlations, and retained for further analyses. Upon graph creation, the degree of each node is calculated and stored. The isolated components of the graphs are identified by a complete uninformed search procedure relying on shortest paths calculation. The hub residues can be also identified. They are highly connected nodes in the network. In packing-based PSN, generally a node is defined a hub if its degree is at least 4,[76] and

dx.doi.org/10.1021/ci300213c | J. Chem. Inf. Model. 2012, 52, 1865–1874

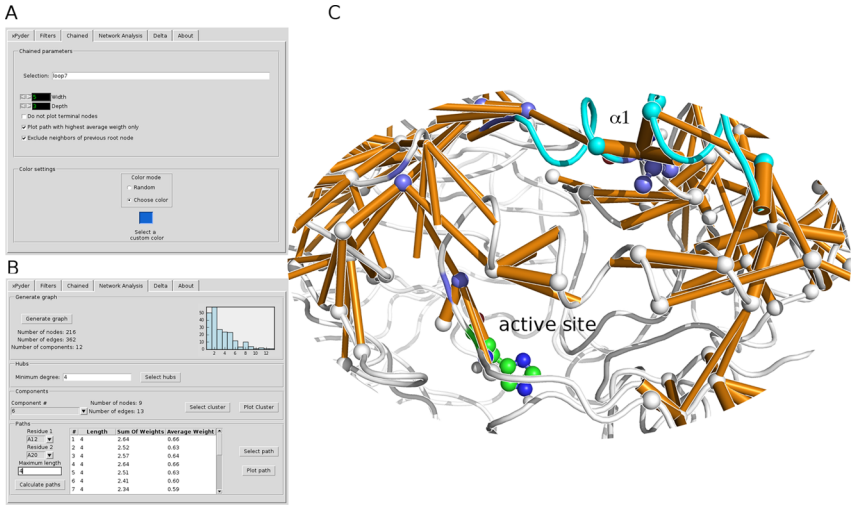


Figure 3. Paths of structural communication. (A) Chained correlation mask of xPyder and its filtering options. (B) Network analysis mask of xPyder and its options. (C) Example of chained correlations calculated using as root residue Val16 (marine stick) of the N-terminal α1-helix of AAP, which is a crucial residue mediating different communication paths between the N-terminal region and two protein domains, as well as communicating to the active site[21] (catalytic hystidine as green stick).

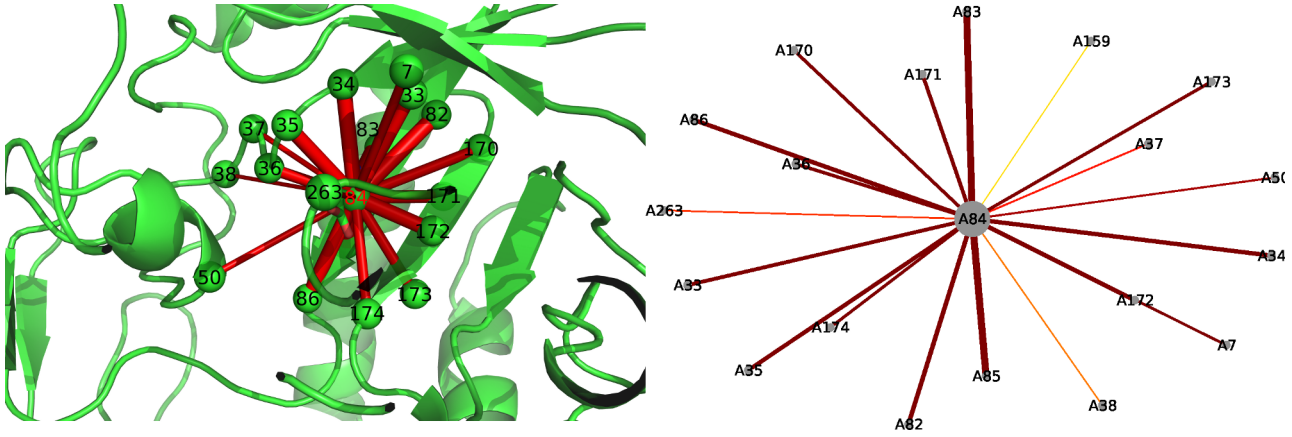


Figure 4. Example of cross-correlation plot. Cross-correlations are plotted as cylinders. The two panels show the cross-correlated motions between residues D84 of a cold-adapted α-amylase and the residues in its surrounding using the filtering options of xPyder. The correlation network is mapped on the 3D structure (left panel) and is shown as a two-dimensional graph (right panel). Some of the residues depicted on the graph in the right panel are hidden by the cartoon representation of the secondary structural elements.

this can be a useful guide for graph analysis of DCCM or other communication metrics. Hubs in the network can play a relevant role because they have been demonstrated in several cases to be residues important for the protein architecture and function.[33,39]

Finally, the user can calculate all the possible paths existing between two nodes of the graph. This is performed by employing a variant of the depth-first search algorithm,[75] in which the history of the visited nodes from a given branch is retained. The searching procedure is carried out so that the same node is not visited more than once to avoid the entrapment in cycles. Moreover, the user may specify the maximum depth of search to adequately tune the completeness/cost ratio of the search.

3. XPYDER PLUGIN

3.1. Overview of xPyder. xPyder guarantees a complete molecular viewer, graphical support, and analysis tools for several steps required in the analysis of communicating or correlated residues, making it a unique tool.

The xPyder PyMOL plugin has been developed specifically to take advantage of the molecular viewing capabilities of the PyMOL suite. In fact, PyMOL is very well suited to generate publication quality pictures of molecular structures, as it allows several advanced rendering options from which xPyder tools can take advantage to carry out detailed analysis and representation of correlated motions, inter- and intermolecular interactions, as well as path of structural communication in biomolecules.

In the current release, the xPyder plugin allows to plot correlations or interactions between residues from a simple plain-text matrix input file (Figure 1, left panel), providing options to filter and refine the plotted data (Figure 1). The filtering options are implemented in a plugin system focused on code reutilization and featuring a full dependency control, so that users can easily implement their own filtering procedures, taking advantage of the functionality provided by other plugins through a specifically conceived programming interface. Options are available to fine-tune the graphical rendering of the results, so that the plugin may be used for easy visualization of complex data and striking imaging for publication, as well as for graph analysis.

The plugin implements different filters on the matrix values to allow the user to decode the information contained in the input

dx.doi.org/10.1021/ci300213c | J. Chem. Inf. Model. 2012, 52, 1865−1874

matrix (Figure 1, right panel). In particular, there is the possibility to filter according to the distance between residues both in the primary sequence and 3D structure, as well as according to the sign of correlations. Filter options according to the secondary structure elements are also available. Moreover, the user can visualize the correlations within a given PyMOL selection or between two selections, thus permitting, for example, for easy display of the correlations between a residue and the rest of a protein or between two chains of a protein complex.

xPyder also includes a module for the comparison between matrices (Figure 2A). It allows for calculation and mapping on the 3D structure of the differences between two matrices according to a provided sequence alignment. This is a useful tool to compare the dynamics patterns between wild-type and mutant variants, as well as between two proteins sharing the same fold or at least comparable through a sequence or a structural alignment. Similar tools were recently applied to the comparison of the coupled motions around the active site in a cold-adapted enzyme and its mesophilic-like mutants, pointing out a modification not only in kinetic and thermodynamics parameters[77] but also in the dynamics fingerprint of the mutant in a mesophilic-like direction[22] (Figure 2B−D).

xPyder features also several options to analyze the graph encoded by the input matrix (Figures 3 and 4). An algorithm for a chained correlations search has been implemented (see Methods for details) to identify the 3D structure of the paths of communication between residues (Figure 3A). The analyses of chains of correlations allows for identification of chains or sequences of residues maximally related to a selected root residues by recursively exploring the edges of the graph. This approach has been recently applied to study the long-range communication paths mediated by the N-terminal α1-helix of a hyperthermophilic acylaminoacyl peptidase (AAP).[21] This helix is known to have a crucial role in protein stability and activity of acylaminoacyl peptidase.[78] In fact, the investigation of chained correlations highlighted a long-range communication from the helix to the catalytic site, which is compromised by deletion of the helix or mutations of hydrophobic residues belonging to the helix itself (Figure 3C). The same deletion is known to alter protein thermal stability and thermal dependence of AAP activity.[78] The same scenario was confirmed by the analysis of the shortest paths of communication between the helix and the catalytic triad carried out by the PSN-DCCM method.[21] Chained correlations can therefore provide useful first hints on the communication routes from different protein sites, which can be then specifically identified for example by PSN approaches.

Graph analysis generally is a useful tool to identify and visualize on a bimolecular structure highly connected residues (hubs), as well as isolated components. In xPyder, the user can visualize and sort according to diverse calculated parameters the paths existing between two residues or two selections, as well as identifying the most interesting communication paths encoded in the matrix (Figure 3B).

A graph-like two-dimensional (2D) scheme can also be provided in output to visualize the networks of plotted correlations in a clear and concise form, as represented in Figure 4 (right panel). This option is available independently on the xPyder mask used for the plotting and filtering options, so that 2D schemes can be achieved both for standard cross-correlations, chained correlations, and network analyses.

3.2. System Requirements and Installation. The main requirement for xPyder is a working installation of PyMOL $\geq$ 1.4 with plugin support, meaning that it will run on most Linux distributions, Windows, and Mac OS X. The plugin also requires the numpy ($\geq$1.5.1) matplotlib ($\geq$1.1) and Tkinter ($\geq$8.4) Python libraries. These modules are currently available in the package management systems of popular Linux distributions, as Ubuntu, Debian, or Fedora, and are otherwise easily installed in most operative systems. Two additional Python modules are required, networkx 1.6 and the Yapsy 1.8 plugin system, which are already included in the installation package. In our installation tests, we found the local Python and PyMOL environment to vary considerably among different operating systems and linux distributions. For example, you may need to install the corresponding Python 2.6 or 2.7 libraries if your PyMOL distribution uses version 2.6 or 2.7, respectively.

To simplify xPyder's installation process, graphical installers are also provided for the major platforms. Support and library files are automatically copied in a user-defined location, and the main xPyder component has to be installed in PyMOL by using the standard plugin handling interface. This allows the plugin to work both on system-wide and personal installations.

3.3. Input Format. The input matrix provided to xPyder must be a square symmetrical matrix. The matrix file format supported by xPyder is a plain-text file consisting of N lines, with N white space-separated floating-point numbers on each line. The entry in the i-th row and the j-th column of the matrix will correspond to the j-th number present on line i.

The dimensions of the matrix correspond to the number N of residues in the analyzed PyMOL object (e.g., for a protein complex of 150 residues in total, the matrix will be of order 150 (150×150)). Given the minimalistic matrix input format, the order of the matrix elements is important. xPyder assumes that for multiprotein complexes the correlation values in the matrix are in the same order as the residues in the corresponding PyMOL structure, i.e., field j of line i of the matrix file describes correlations between residues i and j in the PyMOL object. This is the case in most common applications of xPyder; nevertheless, the user should make sure of the correspondence when using xPyder, as also discussed in details in the User Guide (http://linux.btbs.unimib.it/xpyder/).

The same requirements hold for multichain structures: for example, a protein complex containing chains "A" of 40 residues and chain "B" of 90 residues corresponds to a 130×130 matrix. In this matrix, the rows or columns of indexes 1−40 will refer to residues belonging to chain "A", while the rows or columns of indexes 41−130 will refer to residues of chain "B".

The matrix diagonal contains the correlation values between each residue and itself, and it can be optionally considered for plotting in case the value is meaningful in the metric under investigation. As xPyder is data-agnostic, any correctly formatted matrix may be used: this flexibility is one of the strong points of xPyder, making it an effective and generic plotting tool. Indeed, the simple matrix format has been chosen with versatility in mind, as it can also be easily obtained from custom scripts and general tools such as the MATLAB suite (The MathWorks, Natick, MA) or the NumPy library.[79] Moreover, an xPyder compatible file should be easy obtained from common analysis tools such as those included in Gromacs (www.gromacs.org) or Amber (www.ambermd.org) software suites. Some suggestions about how to obtain a matrix input file suitable for xPyder analyses are available in the User Guide.

For example, the initial data matrix can be obtained with equal ease both from computational simulations and PDB experimental ensembles, making xPyder appealing to both computational and experimental communities. Several freely available

programs have been designed to obtain matricial data both from PDB ensembles and simulation trajectories. The Wordom[80] software is capable of performing a number of analyses on MD-like trajectories (as PSN, DCCM and LMI), which can also be obtained, using the software itself, from PDB ensembles. The Theseus program[81] has been designed to superimpose macromolecular structures from PDB files using the maximum likelihood method and is able to output motion covariance and correlation matrices as well as extracting patterns of correlated motion through principal component analysis. For ease of use, data format conversion tools are available on the xPyder Web site to convert the output files of Wordom and Theseus in the xPyder matrix format.

To plot correlations or any other metric on the protein structure, a 3D structure of a protein or of a macromolecular complex must be loaded in PyMOL, e.g., from a pdb file or any file format supported by PyMOL (see the PyMOL User Guide, http://www.pymolwiki.org). Each protein chain must have consecutive residue numbering but can start at any residue number, and an appropriate chain identifier.

3.4. Plotting and Filtering Options of xPyder. The primary function of the xPyder PyMOL plugin is to visualize information from different metrics of dynamical communication between protein residues directly on the 3D structure. In particular, the values contained in the input matrix are associated to the Cα of the residues on the 3D structure of a protein or of a multimeric complex. Cross-correlations are displayed as colored cylinders (PyMOL CGO objects) connecting Cα atoms with cylinder radius proportional to the entity of the correlation.

Using xPyder first requires loading a DCCM or any other matrix that the user intends to plot on the 3D structure using xPyder's network representation, by selecting the path to the matrix file in the provided input box or by browsing the file system using the provided file selection pop-up (Figure 1, left panel). The user must then define the PyMOL object or selection to which the correlations in the matrix input file refer (Input object name, Figure 1, left panel). By clicking the "Load matrix" button, the matrix file is loaded and processed. xPyder verifies the compatibility between the loaded matrix and the reference object and plots a histogram of the loaded values (Figure 1, left panel). These values might not correspond to those that are going to be plotted, as the matrix filters have not been applied yet. The user can then configure the plotting graphical options (Figure 1, left panel) and the filtering options (Figure 1, right panel).

The filters (Figure 1, right panel) act as cascading masks that filter the input matrix according to structural or dynamics information, which may be calculated by the program or provided as input by the user. Each filtering mask can be independently activated or deactivated, and filtering parameters can be modified to ensure the maximum flexibility. The default filtering plugins, included with the program, cover most of the aspects of correlation analysis: they permit to filter according to distance in sequence or in the 3D structure, so that short- or long-range correlations can be filtered out according to the user necessities. The plotting can also be restricted to a given range of correlation values (in absolute value), so that the lowest or highest values are discarded. This option is useful to filter out weak correlations below a certain threshold, which are likely to be noise and related to uncoupled residues. Correlations can also be filtered on the basis of each residue's flexibility, as defined by the B factor field of the input PDB structure. This can help, for example, in excluding from the correlation analysis those regions that feature the highest experimental uncertainty, but it can also be useful in focusing the analysis on the regions with the highest flexibility during a simulation, thus reinserting in a controlled way the notion of the entity of motions, which is lost in correlation analysis due to normalization. Furthermore, it is a common procedure of analysis tools to replace the B factor field in PDB files with residue-dependent results, and scripts are available for PyMOL that do this on the basis of plain-text data (see http://www.pymolwiki.org/index.php/Color): this filter therefore provides xPyder with fine-tuning capabilities that are extensible through cross-talk with other software.

xPyder can also be made secondary structure-aware both using the PyMOL definition for secondary structure or by loading standard DSSP output files.[82] Our plugin also supports the use of a general secondary structure format, named PDSS (Persistence Degree of Secondary Structures, as detailed in the User Guide), which was designed within the context of molecular dynamics (MD) simulation analyses. The program to obtain a PDSSP file from the xpm output file of the do_dssp utility of GROMACS (www.gromacs.org) is available on the xPyder Web site. Because the PDSS files contain one or more reference secondary structures, the user may choose which one will be used for filtering. It is therefore possible to filter out correlations within secondary structure elements: the user can selectively exclude from the plot the correlations inside the same β-strand, between strands of the same β-sheet, or between residues of the same α-helix (Figure 1, right panel).

Finally, the user can configure a wide range of plotting options (Figure 1, left panel), including the colors and width of the plotted cylinders, to obtain the desired look for the final representation. When performing multiple plots during the same session, the user may choose to change the name of the PyMOL object in which the plot will be stored and can write a log file containing all the plotted correlations in a consistent plain-text format for further analyses. Depending on the type of analysis, the user can optionally plot as spheres the relationship between each residue and itself (i.e., the elements on the diagonal of the loaded matrix), with sphere radius proportional to the relationship entity.

When comparing multiple DCCM or other input matrices, xPyder offers the possibility to compute the difference between matrices (Figure 2), focusing the analysis on the quantitative details of the comparison. The "Delta" analysis of xPyder (Figure 2) allows the user to compute and plot on the 3D structure the difference matrix (or delta matrix), providing the full set of xPyder filtering features to customize the plot. This analysis may prove useful when comparing matrices computed on the same structure (to compare two matrices resulting from two experiments on the same protein, e.g., the correlations from two MD simulations) or on different structures, for example, two homologous proteins. xPyder makes the latter analysis possible by allowing the user to specify an alignment to match the sequences of the two different proteins and therefore the two matrices.

The study of binary interactions, as defined in the matricial representation used by xPyder, can be insightful in detecting relationships between single residue or specific selections but does not tell much about how information is transmitted in the protein structure or about the relationships between distant residues whose motions are not directly correlated. This can be performed by identifying paths of connected residues across the protein structure, using the input matrix as the source of information (Figures 3 and 4). Because the network of relationships is encoded by a matrix, which closely resembles

the graph theory's adjacency matrix, the use of graph theory seems natural to deal with such a problem. With this in mind, a chained correlations identification algorithm, similar to the one of FlexServ,[74] was implemented (see Methods for details) to identify on the 3D structure the paths of communication between residues (Figure 3A). The analyses of chains of correlations allows to identify chains or sequences of residues maximally related to a selected root residues by recursively exploring the edges of the graph. In fact, the algorithm relies on a graph representation of the data of the input matrix, in which nodes correspond to protein residues and weighted edges are taken from the values of the loaded matrix. Given a root residue, the algorithm sorts all the nodes connected to it by edge weight in descending order and selects the first *w* residue in a procedure, which can be iterated using the *w* nodes as new starting points for *d* iterations (see Methods for more details). The algorithm is performed in xPyder on the matrix filtered by the filters plugin system, so that the least meaningful correlations can be left out before exploring the graph.

3.5. Graph Analysis. The graph analysis module can be easily used to explore and analyze the loaded matrix graph (Figures 3 and 4). The graph is calculated upon user demand on the filtered matrix as detailed in the Methods section. A histogram of the degree distribution is presented to the user, so that he can evaluate the shape of the distribution (e.g., understand whether it follows a power law) and the number of connections needed to define hub residues. The number and size of the isolated components of the graph are also identified. The user is able to select on the PyMOL 3D structure the hub residues by providing a minimum degree value and can select and plot the components subgraphs on the 3D structure. Hub residues are interesting because they are often pivotal for the maintenance of the network structure and may correspond to structurally or functionally crucial residues.[33,39] Moreover, their high degree allows them to become critical waypoints for the intersection of the communication pathways.

Isolated components of the graph are collections of nodes for which a path exists between each pair of them and no path exists between every other node of the graph, with each one belonging to the component. In the case of proteins, when using an index describing the residue−residue interactions, they may correspond to local features such as hydrophobic clusters or networks of electrostatic interactions as it was previously shown in several cases of study.[21,22,56,57] Finally, a function able to identify all the pathways existing between residues in the graph is available. This feature is useful to identify pathways of communication between residues, as well as to discriminate between the identified pathways according to cumulative weight, average weight, or path length. The described features can be exploited in a combined fashion to highlight features or communication paths between key residues of a protein, as exemplified in the PSN calculated from MD simulations of E2 enzymes (Figure 5). E2 enzymes have been recently classified in 17 different families,[83] which share the architecture of the catalytic domain. In particular, several of these families account for enzymes that are able to conjugate ubiquitin or ubiquitin-like proteins to the target substrates. The catalytically active E2 presents a conserved HPN (histidine−proline−asparagine) motif in the proximity of the catalytic site. Asn belonging to the HPN motifs along with a conserved tyrosine and serine in the catalytic cleft were shown to be important for the E2-mediated catalysis.[83] Recently, we demonstrated by the PSN-DCCM approach that these residues have an important role also in mediating long-range

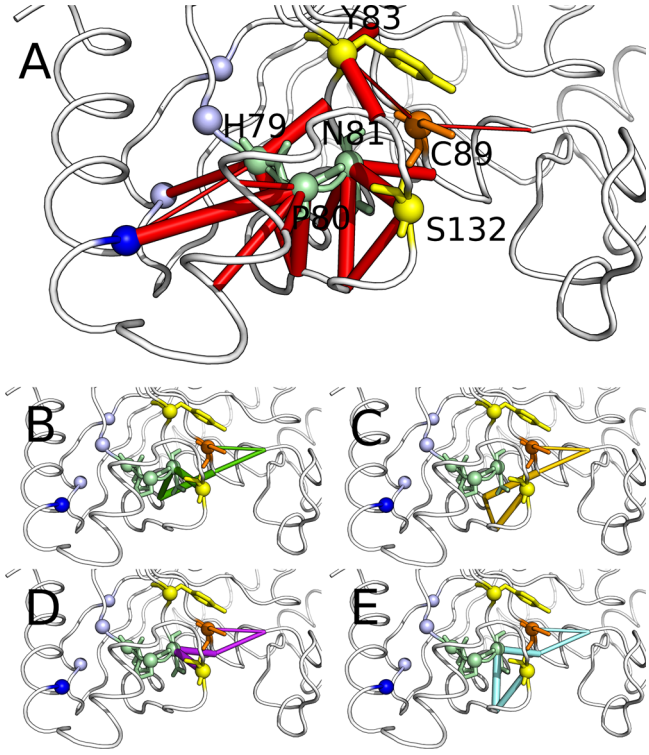


Figure 5. Paths of structural communication in E2 enzymes. Data from PSN analysis carried out by Wordom[80] on a MD simulation[79] of Ubc7 (PDB entry 2UCZ), an E2 enzyme belonging to family 3, were used as input for xPyder analysis with the Network Analysis mask. Catalytically relevant residues are shown in orange, yellow, and green as detailed in the text.

communication from the active site to other protein regions, both structurally and functionally relevant for E2 enzymes of family 3.[84] In Figure 5, a PSN calculated with Wordom[80] was calculated, as an example from a MD simulation of an E2 enzyme of family 3 and plotted on the 3D structure using xPyder. The plotted values are the relative occurrence probability with which an edge was present in the MD trajectory and were filtered by discarding values lower than 0.2. Moreover, only the matrix values involving the key residues mentioned above (in orange, yellow, and green in Figure 5) were plotted. The orange residue is the E2's catalytic Cys, (C89), while Y83 and S134 (in yellow) are the conserved tyrosine and serine mentioned above. The HPN motif is shown in green. The analysis performed with xPyder shows that many of these residues are highly interconnected, with relatively high probability occurrence values for many of the connections. In particular, the HPN residues act as bridges, favoring the communication between residues in this area with several other residues, shown in blue and azure, which are the most connected hubs in the protein (degree 10 and 8, respectively). These residues may thus favor the transmission of information along the protein structure toward the active site. Interestingly, the catalytic cysteine features few low-value edges with the other residues in the area, as also recently demonstrated for the active open states of E2 enzymes of family.[84] The path search algorithm of xPyder has thus been employed to understand if more complex communication paths were possible between C89 and the other residues. For example, no obvious relationships exist between residue S134 and C89; this withstanding, the pathway identification algorithm found 4

paths of length 5 or 6 (Figure 5, lower panels) that connect the two residues with high-weight edges.

3.6. Other Applications of the xPyder Plugin. The general format of the input allows the applicability of xPyder to be extended in principle to any property that can be represented as a network between residues in a conformational ensemble. In other words, xPyder is suitable for analyzing and efficiently visualizing on the 3D structure any kind of pairwise metric, as long as it can be represented as a two-dimensional matrix. The plugin flexibility allows it to be employed successfully to visualize, for example, information from Protein Structure Network (PSN) matrices,[33] Communication Propensity,[36,85] or even DCCM coupled to statistical coupling analysis or other coevolution metrics,[34] apparent pairwise force constants,[86] or simply distance or contact matrices.[87,88] Furthermore, when the latter are constructed by filtering only selected interactions (e.g., salt bridges, aromatic interactions, or hydrogen bonds), xPyder allows the visualization of the 3D localization of the interactions and their relative arrangement, providing invaluable insights into their role in determining the stability and dynamic patterns of the protein architecture. If persistence or energetic values related to the weak intra- and intermolecular interactions are provided from a conformational ensemble, xPyder can be used to visualize not only the networks of intra- or intermolecular interactions but also their strength and persistence in the ensemble, as recently applied to the comparative studies of cold- and warm-adapted enzymes[22,56] or to the identification of tertiary contact stabilizing globular states of intrinsically disordered proteins.[57] An example is shown in Figure 6 for a hyperthemophilic subtilisin.

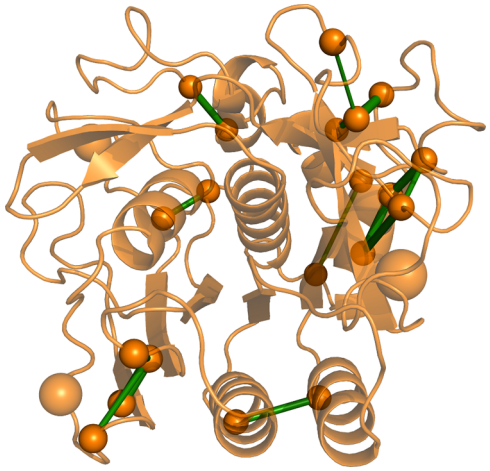


Figure 6. Other applications of the xPyder plugin. Salt bridges and their networks of a subtilisin-like enzyme (pdb entry 1THM) are shown as cylinders connecting Cα atoms of interacting residues. The thickness of the cylinders is proportional to the persistence of the interaction in the MD ensemble.[56] The versatility of the xPyder plugin allows it to plot any kind of data, as long as it is encoded in matrix format. Cα are shown as spheres to better highlight their position; the Ca^{2+} ions are represented as larger spheres.

CONCLUSIONS

The present contribution introduces xPyder, a versatile plugin for PyMOL, which has been designed to plot on the 3D structure and to analyze binary relationships between protein residues encoded in a input matrix. A modular, user-expandable plugin system allows for filtering out the least relevant matrix elements, according to both numerical and structural criteria, taking advantage from state of the art Python scientific libraries. Additional insights can be gained by computing chains of relationships, differences between matrices, and even between different proteins according to a reference alignment, as well as graph analyses. The joint visualization of the dynamic information and 3D localization of the communications and interactions simplify the task of analyzing their relation, contributing to bridging the gap between dynamical and mechanical properties at the molecular level. The plugin is available for PyMOL 1.4 and 1.5 on any supported platform and is offered as Open Source software via the GPL v2 license.

AUTHOR INFORMATION

Corresponding Author

*E-mail: elena.papaleo@unimib.it.

Present Address

[‖]Department of Biology, University of Copenhagen, Ole Maaløes Vej 5, 2200 Copenhagen N, Denmark, elena.papaleo@bio.ku.dk.

Author Contributions

[§]The authors contributed equally to this work.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank Prof. Luca De Gioia, Prof. Richard Lavery, and Prof. Krystyna Zakrzewska for their valuable comments. The authors also thank Matteo Lambrughi, Giulia Renzetti, Matteo Roveda, Alessandro Vitriolo, and Giulia Filippi for testing of xPyder on different biological systems. This research was supported by CASPUR (Consorzio Interuniversitario per le Applicazioni di Supercalcolo per Università e Ricerca) Standard HPC Grant 2011 and 2012, as well as by ISCRA-Cineca (projects HP10COBO3E and HP10CQEI1L) to E.P. M.P. acknowledges funding from the Agence Nationale de la Recherche Grant Expenantio.

REFERENCES

(1) Rueda, M.; Ferrer-Costa, C.; Meyer, T.; Perez, A.; Camps, J.; Hospital, A.; Gelpi, J. L.; Orozco, M. A consensus view of protein dynamics. *Proc. Natl. Acad. Sci. U.S.A.* **2007**, *104*, 796−801.

(2) Markwick, P. R. L.; Bouvignies, G.; Salmon, L.; McCammon, J. A.; Nilges, M.; Blackledge, M. Toward a unified representation of protein structural dynamics in solution. *J. Am. Chem. Soc.* **2009**, *131*, 16968−16975.

(3) Lindorff-Larsen, K.; Best, R. B.; DePristo, M. A.; Dobson, C. M.; Vendruscolo, M. Simultaneous determination of protein structure and dynamics. *Nature* **2005**, *433*, 128−132.

(4) Teilum, K.; Olsen, J. G.; Kragelund, B. B. Functional aspects of protein flexibility. *Cell. Mol. Life Sci.* **2009**, *66*, 2231−2247.

(5) Maguid, S.; Fernandez-Alberti, S.; Echave, J. Evolutionary conservation of protein vibrational dynamics. *Gene* **2008**, *422*, 7−13.

(6) Marcos, E.; Crehuet, R.; Bahar, I. On the conservation of the slow conformational dynamics within the amino acid kinase family: NAGK, the paradigm. *PLoS Comput. Biol.* **2010**, *6*, 14.

(7) Hollup, S. M.; Fuglebakk, E.; Taylor, W. R.; Reuter, N. Exploring the factors determining the dynamics of different protein folds. *Protein Sci.* **2011**, *20*, 197−209.

(8) Law, A. B.; Fuentes, E. J.; Lee, A. L. Conservation of side-chain dynamics within a protein family. *J. Am. Chem. Soc.* **2009**, *131*, 6322−6323.

(9) Maguid, S.; Fernandez-Alberti, S.; Ferrelli, L.; Echave, J. Exploring the common dynamics of homologous proteins. Application to the globin family. *Biophys. J.* **2005**, *89*, 3−13.
(10) Raimondi, F.; Orozco, M.; Fanelli, F. Deciphering the deformation modes associated with function retention and specialization in members of the Ras superfamily. *Structure* **2010**, *18*, 402−14.
(11) Angelova, K.; Felline, A.; Lee, M.; Patel, M.; Puett, D.; Fanelli, F. Conserved amino acids participate in the structure networks deputed to intramolecular communication in the lutropin receptor. *Cell. Mol. Life Sci.* **2011**, *68*, 1227−1239.
(12) Hongbo, S.; Feng, X.; Hairong, H.; Feifei, W.; Qi, W.; Qiang, H.; Honghai, W. Coevolving residues of (beta/alpha)/sub 8/-barrel proteins play roles in stabilizing active site architecture and coordinating protein dynamics. *J. Struct. Biol.* **2008**, *164*, 281−92.
(13) Kornev, A. P.; Taylor, S. S. Defining the conserved internal architecture of a protein kinase. *Biochim. Biophys. Acta* **2010**, *1804*, 440−444.
(14) Bocahut, A.; Bernad, S.; Sebban, P.; Sacquin-Mora, S. Frontier residues lining globin internal cavities present specific mechanical properties. *J. Am. Chem. Soc.* **2011**, *133*, 8753−8761.
(15) Echave, J.; Fernandez, F. M. A perturbative view of protein structural variation. *Proteins* **2010**, *78*, 173−180.
(16) Zheng, W. J.; Brooks, B. R.; Thirumalai, D. Low-frequency normal modes that describe allosteric transitions in biological nanomachines are robust to sequence variations. *Proc. Natl. Acad. Sci. U.S.A.* **2006**, *103*, 7664−7669.
(17) Zheng, W.; Brooks, B. R.; Thirumalai, D. Allosteric transitions in the chaperonin GroEL are captured by a dominant normal mode that is most robust to sequence variations. *Biophys. J.* **2007**, *93*, 2289−2299.
(18) Papaleo, E.; Pasi, M.; Riccardi, L.; Sambi, I.; Fantucci, P.; De Gioia, L. Protein flexibility in psychrophilic and mesophilic trypsins. Evidence of evolutionary conservation of protein dynamics in trypsin-like serine-proteases. *FEBS Lett.* **2008**, *582*, 1008−1018.
(19) Wrabl, J. O.; Gu, J. N.; Liu, T.; Schrank, T. P.; Whitten, S. T.; Hilser, V. J. The role of protein conformational fluctuations in allostery, function, and evolution. *Biophys. Chem.* **2011**, *159*, 129−141.
(20) Moorthy, B. S.; Anand, G. S. Multistate allostery in response regulators: Phosphorylation and mutagenesis activate RegA via alternate modes. *J. Mol. Biol.* **2012**, *417*, 468−487.
(21) Papaleo, E.; Renzetti, G.; Tiberti, M. Mechanisms of intramolecular communication in a hyperthermophilic acylaminoacyl peptidase: a molecular dynamics investigation. *PLoS One* **2012**, *7*, e35686.
(22) Papaleo, E.; Pasi, M.; Tiberti, M.; De Gioia, L. Molecular dynamics of mesophilic-like mutants of a cold-adapted enzyme: Insights into distal effects induced by the mutations. *PLoS One* **2011**, *6*, e24214.
(23) Glembo, T. J.; Farrell, D. W.; Gerek, Z. N.; Thorpe, M. F.; Ozkan, S. B. Collective dynamics differentiates functional divergence in protein evolution. *PLoS Comput. Biol.* **2012**, *8*, e1002428.
(24) Dastidar, S. G.; Raghunathan, D.; Nicholson, J.; Hupp, T. R.; Lane, D. P.; Verma, C. S. Chemical states of the N-terminal "lid" of MDM2 regulate p53 binding simulations reveal complexities of modulation. *Cell Cycle* **2011**, *10*, 82−89.
(25) Lovera, S.; Sutto, L.; Boubeva, R.; Scapozza, L.; Dolker, N.; Gervasio, F. L. The different flexibility of c-Src and c-Abl kinases regulates the accessibility of a druggable inactive conformation. *J. Am. Chem. Soc.* **2012**, *134*, 2496−2499.
(26) Karplus, M.; Kuriyan, J. Molecular dynamics and protein function. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 6679−6685.
(27) Klepeis, J. L.; Lindorff-Larsen, K.; Dror, R. O.; Shaw, D. E. Long-timescale molecular dynamics simulations of protein structure and function. *Curr. Op. Struct. Biol.* **2009**, *19*, 120−127.
(28) Dodson, G. G.; Lane, D. P.; Verma, C. S. Molecular simulations of protein dynamics: New windows on mechanisms in biology. *Embo Rep.* **2008**, *9*, 144−150.
(29) Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; shaw, D. E. Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins: Struct., Funct., Bioinf.* **2010**, *78*, 1950−1958.
(30) Lindorff-Larsen, K.; Maragakis, P.; Piana, S.; Eastwood, M. P.; Dror, R. O.; Shaw, D. E. Systematic validation of protein force fields against experimental data. *PLoS One* **2012**, *7*, e32131.
(31) Boede, C.; Kovacs, I. A.; Szalay, M. S.; Palotai, R.; Korcsmaros, T.; Csermely, P. Network analysis of protein dynamics. *Febs Lett.* **2007**, *581*, 2776−2782.
(32) Csermely, P.; Palotai, R.; Nussinov, R. Induced fit, conformational selection and independent dynamic segments: an extended view of binding events. *Trends Biochem. Sci.* **2010**, *35*, 539−546.
(33) Vishveshwara, S.; Ghosh, A.; Hansia, P. Intra and inter-molecular communications through protein structure network. *Curr. Prot. Pept. Sci.* **2009**, *10*, 146−160.
(34) Estabrook, R. A.; Luo, J.; Purdy, M. M.; Sharma, V.; Weakliem, P.; Bruice, T. C.; Reich, N. O. Statistical colevolution analysis and molecular dynamics: Identification of amino acid pairs essential for catalysis. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 994−999.
(35) Vijayabaskar, M. S.; Vishveshwara, S. Interaction energy based protein structure networks. *Biophys. J.* **2010**, *99*, 3704−3715.
(36) Morra, G.; Verkhivker, G.; Colombo, G. Modeling signal propagation mechanisms and ligand-based conformational dynamics of the Hsp90 molecular chaperone full-length dimer. *PLoS Comput. Biol.* **2009**, *5*, e1000323.
(37) Stacklies, W.; Xia, F.; Grater, F. Dynamic allostery in the methionine repressor revealed by force distribution analysis. *PLoS Comput. Biol.* **2009**, *5*, e1000574.
(38) Ghosh, A.; Vishveshwara, S. A study of communication pathways in methionyl-tRNA synthetase by molecular dynamics simulations and structure network analysis. *Proc. Natl. Acad.Sci. USA.* **2007**, *104*, 15711−15716.
(39) Fanelli, F.; Felline, A. Dimerization and ligand binding affect the structure network of A(2A) adenosine receptor. *Biochim. Biophys. Acta* **2011**, *1808*, 1256−1266.
(40) Uversky, V. N.; Roman, A.; Oldfield, C. J.; Dunker, A. K. Protein intrinsic disorder and human papillomaviruses: increased amount of disorder in E6 and E7 oncoproteins from high risk HPVs. *J. Prot. Res.* **2006**, *5*, 1829−1842.
(41) Armenta-Medina, D.; Perez-Rueda, E.; Segovia, L. Identification of functional motions in the adenylate kinase (ADK) protein family by computational hybrid approaches. *Proteins: Struct., Funct., Bioinf.* **2011**, *79*, 1662−1671.
(42) Ackerman, S. H.; Gatti, D. L. The contribution of coevolving residues to the stability of KDO8P synthase. *PLoS One* **2011**, *6*, e17459.
(43) Silvestre-Ryan, J.; Lin, Y. C.; Chu, J. W. "Fluctuograms" reveal the intermittent intra-protein communication in subtilisin Carlsberg and correlate mechanical coupling with co-evolution. *PLoS Comput. Biol.* **2011**, *7*, e1002023.
(44) Tsai, C. J.; del Sol, A.; Nussinov, R. Allostery: Absence of a change in shape does not imply that allostery is not at play. *J. Mol. Biol.* **2008**, *378*, 1−11.
(45) Gunasekaran, K.; Ma, B. Y.; Nussinov, R. Is allostery an intrinsic property of all dynamic proteins? *Proteins: Struct., Funct., Bioinf.* **2004**, *57*, 433−443.
(46) Cui, Q.; Karplus, M. Allostery and cooperativity revisited. *Protein Sci.* **2008**, *17*, 1295−1307.
(47) Swain, J. F.; Gierasch, L. M. The changing landscape of protein allostery. *Curr. Op. Struct. Biol.* **2006**, *16*, 102−108.
(48) Hunenberger, P. H.; Mark, A. E.; Vangunsteren, W. F. Fluctuation and cross-correlation analysis of protein motions observed in nanosecond molecular dynamics simulations. *J. Mol. Biol.* **1995**, *252*, 492−503.
(49) Bhardwaj, N.; Abyzov, A.; Clarke, D.; Shou, C.; Gerstein, M. B. Integration of protein motions with molecular networks reveals different mechanisms for permanent and transient interactions. *Protein Sci.* **2011**, *20*, 1745−1754.
(50) Doncheva, N. T.; Klein, K.; Domingues, F. S.; Albrecht, M. Analyzing and visualizing residue networks of protein structures. *Trends Biochem. Sci.* **2011**, *36*, 179−182.

(51) Doncheva, N. T.; Assenov, Y.; Domingues, F. S.; Albrecht, M. Topological analysis and interactive visualization of biological networks and protein structures. *Nature Protocols* **2012**, *7*, 670−685.
(52) Martin, A. J.; Vidotto, M.; Boscariol, F.; Di Domenico, T.; Walsh, I.; Tosatto, S. C. RING: Networking interacting residues, evolutionary information and energetics in protein structures. *Bioinformatics* **2011**, *27*, 2003−2005.
(53) Ahmad, S.; Keskin, O.; Mizuguchi, K.; Sarai, A.; Nussinov, R. CCRXP: Exploring clusters of conserved residues in protein structures. *Nucleic Acids Res.* **2010**, *38*, W398−W401.
(54) Durrant, J. D.; McCammon, J. A. HBonanza: A computer algorithm for molecular-dynamics-trajectory hydrogen-bond analysis. *J. Mol. Graph. Modell.* **2011**, *31*, 5−9.
(55) V., M. S.; Vidya, N.; Saraswathi, V. GraProStr − Graphs of protein structures: A tool for constructing the graphs and generating graph parameters for protein structures. *Open Bioinf. J.* **2011**, *5*, 53−58.
(56) Tiberti, M.; Papaleo, E. Dynamic properties of extremophilic subtilisin-like serine-proteases. *J. Struct. Biol.* **2011**, *174*, 69−83.
(57) Arrigoni, A.; Grillo, B.; Vitriolo, A.; Gioia, L. D.; Papaleo, E. C-terminal acidic domain of ubiquitin-conjugating enzymes: A multi-functional conserved intrinsically disordered domain in family 3 of E2 enzymes. *J. Struct. Biol.* **2012**, *178*, 245−259.
(58) Sanner, M. F. Python: A programming language for software integration and development. *J. Mol. Graph. Modell.* **1999**, *17*, 57−61.
(59) Vehlow, C.; Stehr, H.; Winkelmann, M.; Duarte, J. M.; Petzold, L.; Dinse, J.; Lappe, M. CMView: Interactive contact map visualization and analysis. *Bioinformatics* **2011**, *27*, 1573−1574.
(60) Lua, R. C.; Lichtarge, O. PyETV: A PyMOL evolutionary trace viewer to analyze functional site predictions in protein complexes. *Bioinformatics* **2010**, *26*, 2981−2982.
(61) Rother, K.; Hildebrand, P. W.; Goede, A.; Gruening, B.; Preissner, R. Voronoia: Analyzing packing in protein structures. *Nucleic Acids Res.* **2009**, *37*, D393−D395.
(62) Petrek, M.; Otyepka, M.; Banas, P.; Kosinova, P.; Koca, J.; Damborsky, J. CAVER: A new tool to explore routes from protein clefts, pockets and cavities. *BMC Bioinf.* **2006**, *7*, 316.
(63) Hagelueken, G.; Ward, R.; Naismith, J. H.; Schiemann, O. MtsslWizard: In silico spin-labeling and generation of distance distributions in PyMOL. *Appl. Magn. Reson.* **2012**, *42*, 377−391.
(64) Dantzer, J.; Moad, C.; Heiland, R.; Mooney, S. MutDB services: Interactive structural analysis of mutation data. *Nucleic Acids Res.* **2005**, *33*, W311−W314.
(65) Ordog, R. PyDeT, a PyMOL plug-in for visualizing geometric concepts around proteins. *Bioinformation* **2008**, *2*, 346−347.
(66) Bramucci, E.; Paiardini, A.; Bossa, F.; Pascarella, S. PyMod: Sequence similarity searches, multiple sequence-structure alignments, and homology modeling within PyMOL. *Bmc Bioinf.* **2012**, *13* (Suppl 4), S2.
(67) Bakan, A.; Meireles, L. M.; Bahar, I. ProDy: Protein dynamics inferred from theory and experiments. *Bioinformatics* **2011**, *27*, 1575−1577.
(68) Cabrera, Á.; Gil-Redondo, R.; Perona, A.; Gago, F.; Morreale, A. VSDMIP 1.5: An automated structure- and ligand-based virtual screening platform with a PyMOL graphical user interface. *J. Comput.-Aided Mol. Des.* **2011**, *25*, 813−24.
(69) Lill, M. A.; Danielson, M. L. Computer-aided drug design platform using PyMOL. *J. Comput.-Aided Mol. Des.* **2011**, *25*, 13−19.
(70) Seeliger, D.; de Groot, B. L. Ligand docking and binding site analysis with PyMOL and Autodock/Vina. *J. Comput.-Aided Mol. Des.* **2010**, *24*, 417−422.
(71) Hodis, E.; Schreiber, G.; Rother, K.; Sussman, J. L. eMovie: A storyboard-based tool for making molecular movies. *Trends Biochem. Sci.* **2007**, *32*, 199−204.
(72) Baugh, E. H.; Lyskov, S.; Weitzner, B. D.; Gray, J. J. Real-Time PyMOL visualization for Rosetta and PyRosetta. *PLoS One* **2011**, *6*, 5.
(73) Lange, O. F.; Grubmuller, H. Generalized correlation for biomolecular dynamics. *Proteins: Struct., Funct., Bioinf.* **2006**, *62*, 1053−1061.
(74) Camps, J.; Carrillo, O.; Emperador, A.; Orellana, L.; Hospital, A.; Rueda, M.; Cicin-Sain, D.; D'Abramo, M.; Gelpi, J. L.; Orozco, M. FlexServ: An integrated tool for the analysis of protein flexibility. *Bioinformatics* **2009**, *25*, 1709−1710.
(75) Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Clifford, S. *Introduction to Algorithms*, 3rd ed.; **2009**; Vol. *1*, p 1292.
(76) Brinda, K. V.; Vishveshwara, S. A network representation of protein structures: Implications for protein stability. *Biophys. J.* **2005**, *89*, 4159−4170.
(77) D'Amico, S.; Gerday, C.; Feller, G. Temperature adaptation of proteins: Engineering mesophilic-like activity and stability in a cold-adapted alpha-amylase. *J. Mol. Biol.* **2003**, *332*, 981−988.
(78) Zhang, Z. M.; Zheng, B. S.; Wang, Y. P.; Chen, Y. Q.; Manco, G.; Feng, Y. The conserved N-terminal helix of acylpeptide hydrolase from archaeon Aeropyrum pernix K1 is important for its hyperthermophilic activity. *Biochim. Biophys. Acta* **2008**, *1784*, 1176−1183.
(79) Oliphant, T. E. *Guide to NumPy*; Brigham Young University: Provo, UT, 2006.
(80) Seeber, M.; Felline, A.; Raimondi, F.; Muff, S.; Friedman, R.; Rao, F.; Caflisch, A.; Fanelli, F. Wordom: A user-friendly program for the analysis of molecular structures, trajectories, and free energy surfaces. *J. Comput. Chem.* **2011**, *32*, 1183−1194.
(81) Theobald, D. L.; Wuttke, D. S. Accurate structural correlations from maximum likelihood superpositions. *PLoS Comput. Biol.* **2008**, *4*, e43.
(82) Kabsch, W.; Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **1983**, *22*, 2577−2637.
(83) Michelle, C.; Vourc'h, P.; Mignon, L.; Andres, C. R. What was the set of ubiquitin and ubiquitin-like conjugating enzymes in the eukaryote common ancestor? *J. Mol. Evol.* **2009**, *68*, 616−628.
(84) Papaleo, E.; Lindorff-Larsen, K.; De Gioia, L. Paths of long-range communication in the E2 enzymes of family 3: A molecular dynamics investigation, *Phys. Chem. Chem. Phys.*, DOI: 10.1039/C2CP41224A
(85) Chennubhotla, C.; Bahar, I. Signal propagation in proteins and relation to equilibrium fluctuations. *PLoS Comput. Biol.* **2007**, *3*, 1716−1726.
(86) Sacquin-Mora, S.; Lavery, R. Investigating the local flexibility of functional residues in hemoproteins. *Biophys. J.* **2006**, *90*, 2706−2717.
(87) Park, K.; Vendruscolo, M.; Domany, E. Toward an energy function for the contact map representation of proteins. *Proteins: Struct., Funct., Bioinf.* **2000**, *40*, 237−248.
(88) Caprara, A.; Carr, R.; Istrail, S.; Lancia, G.; Walenz, B. 1001 optimal PDB structure alignments: Integer programming methods for finding the maximum contact map overlap. *J. Comput. Biol.* **2004**, *11*, 27−52.