BagPype: A Python package for the construction of atomistic, energy-weighted graphs from biomolecular structures

Florian J. Song,[†,‡] Mauricio Barahona,[¶,‡] and Sophia N. Yaliraki[*,†,‡]

†*Department of Chemistry, Imperial College London, London SW7 2AZ, UK*

‡*Institute of Chemical Biology, Imperial College London, London SW7 2AZ, UK*

¶*Department of Mathematics, Imperial College London, London SW7 2AZ, UK*

E-mail: s.yaliraki@imperial.ac.uk

Abstract

Atomistic, energy-weighted graphs of biomolecular structures allow for versatile and efficient modelling of their properties whilst keeping physico-chemical detail. Starting only with *a priori* knowledge of the spatial arrangement of individual atoms obtained from structural files available at the Protein Data Bank (PDB), we present a multi-step pipeline leading to an atomistic energy-weighted graph with individual atoms as nodes and chemical interactions as edges. Whilst most graph approaches only consider strong interactions and typically only at the residue level, an advantage of our methodology lies in the inclusion of weaker interactions, such as hydrogen bonds, electrostatics, hydrophobic interactions and π-π stacking interactions in DNA. The latter enable the study of nucleic acids and their complexes with proteins. In addition, we provide an implementation of the framework in the Python programming language, which is made available under the GNU General Public License v3.0 at `https://github.com/`

`yalirakilab/BagPype`. The computational efficiency of the programme is shown by obtaining wall-clock timing data for over 50,000 experimentally obtained structures spanning most of the PDB. We find that our implementation scales as a slow-growing second order polynomial, where even the largest structures consisting of more than 60,000 residues can be processed in only a few minutes on a standard desktop computer. Finally, a case study of the well-studied *lac* operon repressor protein-DNA complex, comprising of 10,937 atoms, showcases aspects of the methodology using a dynamics-based graph clustering technique, which has been previously applied successfully to elucidate protein rigidity and multi-scale organisation. The graphs obtained by the approach presented here can be combined with any method that uses graph theoretic or network scientific information.

Introduction

Biomolecular structures have often been studied from a graph theoretical perspective, but almost exclusively at a coarse-grained (CG) level. For example, residue interaction networks[1–7] or elastic network models (ENMs),[8,9] have been successful in describing protein properties such as large-scale, biologically relevant structural dynamics of proteins[10] at a low computational cost. Other graph theoretic perspectives on proteins include Markov State Models[11] and Transition Networks,[12] where networks of particular conformations, usually taken from molecular dynamics simulations, are used to study various aspects of long-term behaviour.

Despite the wealth of methods based on CG models, there are yet comparatively few approaches using atomistically detailed graphs, where nodes correspond exclusively to single atoms rather than residues. Though nowadays mostly used in their CG variants, elastic network models were originally proposed at the atomistic level as networks of Hookean springs.[13] More recent developments have shown great promise in areas such as deciphering protein rigidity,[14,15] binding site prediction[16] as well as characterisation of bioactivity.[17,18] The edges have been defined in different ways based on varying functions of inter-atomic distances and cut-offs,[19,20] while approaches combining CG models with atomistic details have also been proposed.[21–23] The most recent approaches (at both CG or atomistic levels of detail) usually also combine graph-based methodology with some form of machine learning algorithms, in order to achieve both detailed but also computationally inexpensive models.[16]

Many atomistic approaches rely on covalent bonds exclusively[18,24] and weak interactions or bonds are often ignored due to the separation of scales. However, it is the latter kind of bonds, such as hydrogen bonds, hydrophobic interactions and π-π stacking interactions, that are pivotal in most processes, such as ligand binding,[25] allostery,[26] multimer interactions and protein-protein interactions to name a few.

We have recently proposed a methodology for constructing fully atomistic, energy-weighted graphs on proteins,[27–29] where a node corresponds to an atom and the edges are weighted by energetic rather than distance-based values of both strong and weak interactions. The

atomistic nature of the resulting graphs allowed us to not only retain the full multiscale resolution provided by the original structure,[30,31] but also enabled the exploration of amino acid side-chain interactions that would otherwise be lost through coarse-graining.[26] However, this approach was limited to proteins and relied on the programme Floppy Inclusion and Rigid Substructure Topography (FIRST)[14,15,32] for the detection of edges. Here, we provide a generalisation of the methodology that is applicable to both protein and protein-deoxyribonucleic acid (DNA) complexes and is modified and extended in several additional ways: it introduces a novel approach to detecting hydrophobic edges by using a recently developed nonlinear geometric graph sparsification technique;[33] it introduces π-π stacking interactions and electrostatic backbone interactions for DNA; its implementation called `BagPype` (Biochemical atomistic graph construction software in Python for proteins etc) is standalone, open-source and written in Python, a popular, modern high-level programming language;[34] it is computationally efficient so that a graph of a biomolecular structure can be obtained in a few minutes on a current, standard laptop or desktop.

Beyond obtaining a much more detailed depiction of the biomolecule, there are many more advantages to including weak bonds at atomistic resolution. In order for proteins to fulfil many of their purposes, especially those related to genetic activity such as transcription, packaging, rearrangement, replication and repair,[35] they often bind to DNA to form protein-DNA complexes. Whilst previous work has been limited to parsing proteins or DNA in isolation, we are now enabling the study of protein-DNA complexes through the inclusion of DNA-specific types of chemical interactions, in particular π-π stacking interactions and DNA backbone electrostatic interactions. The interplay between proteins and nucleic acids is essential to many intracellular processes responsible for maintaining and expressing genetic information.[36,37] Insight into the formation and activity mechanisms as well as the ability to predict these protein-DNA interactions would provide useful guidance for many areas of application, particularly drug discovery.[38,39]

Atomistic graphs that retain physico-chemical detail have been shown to reveal biophysi-

cal properties at multiple resolutions, allosteric properties and communication. This general methodology with an open-source Python implementation can be interfaced with any graph or network approach for further study.

Methodology

Overview

We describe a systematic method for the representation of biomolecules as atomistic, energy-weighted graphs, together with a general-purpose, standalone implementation `BagPype` in Python[34] (version 3.6 and newer), a widely-used high-level programming language. The approach introduced herein is however entirely independent of language-specific features and can therefore be implemented in any current established general-purpose/scientific programming language. A complete version of the code can be obtained at `https://github.com/yalirakilab/BagPype`. Note that since the programme described here runs very efficiently (as demonstrated later on), we have forgone using any multi-processing functionalities. Since the entire programme runs single-threaded, this methodology is highly suitable for *in silico* high-throughput analysis.

A *graph* can be mathematically defined by a set of nodes together with a set of edges consisting of pairs of nodes that are connected. With this methodology, the aim to is construct an *atomistic, weighted* graph and so the set of nodes is simply the set of all individual atoms present in the biomolecule. This can be obtained from an experimentally determined and/or computationally optimised structure of the molecule, alongside the spatial coordinates in three dimensions, the names, residues and other information. The challenge now lies in computing the set of edges given the information from the structure as well as defining a weighting function for every edge, the latter of which will yield a weighted graph and allows for the distinction of the various types of bonds and interactions included. Fundamentally, the graph construction problem can be split into two sub-tasks: The detection of a bond or

Table 1: Table summarising type of bonds and potentials used.

Bond/interaction type	Edge detection	Weighting potential
Covalent bonds	Chemical Component Dictionary[40] + distance constraints[41,42]	Bond dissociation energies[43]
Hydrogen bonds	Distance & angle constraints[14] + energetic cutoff	Modified Mayo potential (DREIDING)[44,45]
Hydrophobic interactions	Set of constraints[14] + Relaxed minimum spanning tree[33,46]	Hydrophobic potential of mean force[47]
Electrostatic interactions	Only for specific ions/ligands as defined by `LINK` entries in PDB file	Coulomb potential with partial charges from OPLS-AA force field[48]
π-π stacking interactions	energetic cutoff + equal division across ring moeities	Potential combining vdW and electrostatic contribution[49,50]

interaction between any given pair of atoms and the weighting of the bond/interaction. This can be seen in the second and fourth columns of figure 1, respectively.

As mentioned in the introduction, we not only consider covalent bonds, but also include weaker types of bonds and interactions. Therefore, the edge-weight function will not attain a single but rather several functional forms depending on the type of bond or interaction in question. In addition, the edge detection will also depend on the bond type. In table 1, we have provided a comprehensive overview of all bond/interaction types alongside their respective bond detection and edge weighting methods. For a more comprehensive treatise of each potential, see the supplementary information.

In contrast, figure 1 presents a visual guide through the entire process starting with atomistic three-dimensional (3D) coordinates and finishing with an atomistic, energy-weighted graph and the corresponding mathematical representation, e.g. its adjacency matrix. Two example systems are shown, of which the first is caffeine. Its simple structure helps with understanding the process at the level of individual atoms and bonds. On the other hand, the *lac* repressor DNA binding domain shows the process applied to a much bigger molecule, leading to a large atomistic graph consisting of roughly 1500 atoms.

In the following, we first describe the input file handling, followed by a detailed discussion

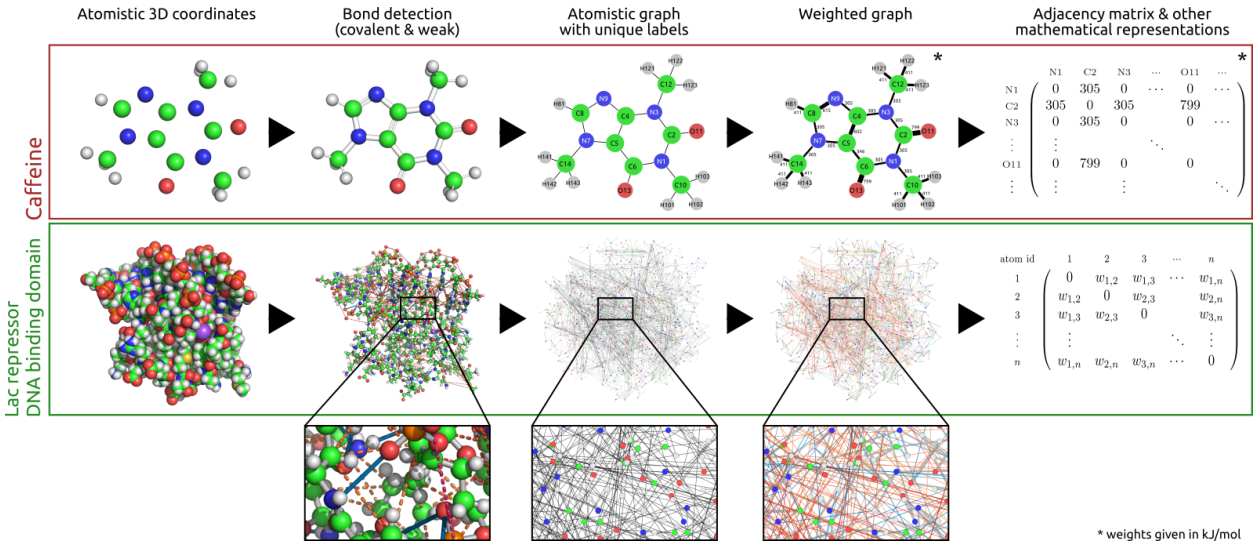

Figure 1: Representative visualisation of the atomistic graph construction process. Two example systems are given: the small molecule caffeine and the *lac* repressor DNA binding domain which is discussed in this work. Starting with three-dimensional coordinates of individual atoms, the first step is to detect covalent bonds as well as weak interactions. This is then converted to an unweighted graph where the nodes are all the atoms and the edges correspond to their interactions. Finally, the edges are weighted using information from atomistic interaction potentials. The weighted graph encodes the physico-chemical detail of the biomolecule and can be converted to an adjacency matrix or other useful graph-theoretical representations for further study. Whilst the simplicity of caffeine's molecular structure helps with understanding the process at the atomistic level, the second example shows the graph construction applied to a much bigger protein-DNA complex, namely the DNA binding domain of the *lac* repressor protein. Atoms are coloured according to their type consistently throughout the figure, while the thickness of the edge scales with the edge weight. For the *lac* repressor, edges are coloured according to their interaction types (covalent bonds in grey, hydrogen bonds in blue and hydrophobic interactions in orange).

of each bond type considered in our graph construction. This is representative of the rough order of a typical run of the program.

Input file cleanup and parsing

The sole input of a standard run in `BagPype` in the implementation here is a structural file usually obtained from the Protein Data Bank[51] (PDB). Note that any input file has to be in the PDB format (version 3.30) which is the international standard for biomolecular structures (for details, see `http://www.wwpdb.org/documentation/file-format`). Functionality to

include further file formats will be part of a future release.

Firstly, a typical execution involves some initial cleaning of the file, such as stripping the file of `ANISOU` entries and any unwanted ligands/solvents or combining NMR models. Furthermore, it is possible to select the alternate location indicator, which is present only if an atom is provided in more than one location. We have also implemented a dedicated filtering functionality, with which unwanted entries in the file can be stripped according to a range of fields, such as residue name or sequence number, element symbol or atom serial number.

After the cleanup, the biological units are generated from the asymmetric unit described by the `REMARK 350` and `BIOMT` fields of the PDB file. To this end, we have improved and incorporated a Python script[52] from the (defunct at time of writing) MakeMultimer server. Then, the programme determines whether hydrogen atoms are already present in the structure. If this is not the case, the third-party software package Reduce[53](available at `http://kinemage.biochem.duke.edu/software/reduce.php` is used to add missing hydrogen atoms.

Finally, this is followed by a parsing step, which extracts the three-dimensional coordinates of all atoms and other useful information such as residue and chain identifiers, which is then utilised by the various edge detection methods described in the following.

Note that whilst the default functionality has been set so that most files are parsed reasonably, this may not always be the case, especially when input files are pre-processed rather than input unchanged from the PDB. Therefore, any functionality mentioned above may be switched on or off at the user's discretion to provide greater flexibility with regards to the input file's particular information and layout.

Detection and weighting of bonds and interactions

Whilst each type of bond/interaction is implemented through a specific, independent routine with a wide variety of components, there is one commonality shared between all. Every

detection routine operates by looping through all atoms of the structure once. For each atom, we consider only atoms in the vicinity (closer than 9Å) as potential bonds/interactions. For this, all pair-wise distances are only computed once and each atoms close neighbours are stored by the program. This way, we are able to save a tremendous amount of computation time which we show later on, achieving a near-linearly scaling algorithm as opposed to quadratic time if one were to consider all possible atom pairs.

Covalent bonds Covalent bonds play a fundamental role in the structural layout of any biomolecule and their topological importance has been studied extensively. The covalent bonding structure of compounds and residues can be obtained from various databases. In our implementation, we make use of the Chemical Component Dictionary,[40] which contains all residues and small molecule components found in the PDB. This database is particularly suited for the study of proteins as well as the large variety of ligands, other small molecules and ions, as it retains the naming conventions of the PDB, i.e. each standard residue or small molecule is distinguishable through a three letter code (e.g. `GLY` = glycine, `CFF` = caffeine). In case of non-standard residues that do not appear in the CCD as may be the case for unpublished structures, our implementation provides a functionality for manually adding information on covalent bonds for the specific residue.

In contrast to the *intra*-residue bonds described above, *inter*-residue bonds are determined differently. In general, only peptide bonds (C-N), nucleic bonds (O-P) as well as disulfide bridges (S-S) are allowed. Note that while most *inter*-residue bonds will be of the afore-mentioned types, we have also included a functionality for forcibly including them via a `LINK` entry in the PDB file, as a way to circumvent these restrictions. To determine whether two atoms of different residues are covalently bonded, we utilise atom-specific covalent radii derived by Pyykkö and Atsumi,[42] where a bond is deemed to be present if the atomic distance is no more than the sum of the constituent atomic radii plus some tolerance.

To assign strengths of interactions to the covalent bonds, we use the bond dissociation

energy as reported in the comprehensive work in Ref,[43] accounting for bond multiplicities. In moieties which exhibit electron delocalisation such as aromatic rings (automatically detected[54]) as well as carboxylic acid and guanidin groups (in standard amino acid residues), the bond weights are averaged across the moiety to account for delocalisation effects.

Through this approach, differences in bond energy arising from different molecules and moieties as well as dependence on the position within the molecule are neglected in favour of generality. Instead, we are able to incorporate into the full atomistic graph of the biomolecule or complex any non-standard structure (e.g. small molecule ligands) in addition to standard amino acids and nucleic acids, thus dramatically increasing the general-purpose nature of our work.

Hydrogen bonds The importance of hydrogen bonds within biomolecules and particularly their contribution to the stability of proteins and ligand binding is well known.[14,25,55–57]

In this work, we adopt a method introduced by Jacobs et al.[14] in the software package FIRST, which detects hydrogen bonds and salt bridges through a set of geometric constraints as well as an energy function[44] stemming from the explicit hydrogen bond component of the DREIDING force field introduced by Mayo et al.,[45] corresponding to equation S2 in the supplementary material.

In order to ensure that only physically realistic arrangements of atoms are considered, the following list of constraints is imposed to determine a set of possible hydrogen bonds: (i) Both donor and acceptor atoms have to be either a nitrogen, an oxygen or a sulfur. (ii) The distance between donor and acceptor has to be less than 5Å. (iii) The distance between hydrogen and acceptor has to be less than 4Å. (iv) The angle formed by donor, hydrogen and acceptor has to be between 100° and 180°. Essentially, these restrictions, consisting mainly of geometric bounds, provide a preliminary filter that discards any unrealistic constellations before any further calculations are done.

Whilst FIRST only uses the Mayo potential to detect hydrogen bonds, we additionally

use it to calculate the bond energy. To this end, every nitrogen, oxygen and sulfur atom of the structure is assigned a so-called hydrogen bond status which allows for the distinction between hydrogen bonds and salt bridges as well as providing the necessary parameters for S2. Finally, candidate hydrogen bonds are then subject to a fixed cut-off parameter (by default at -0.01kcal/mol).

Electrostatic interactions Though generally neglected, we include some essential types of electrostatic interactions in this work, as they are crucial to the structure and dynamics of proteins and DNA: salt bridges, electrostatic interactions between ions/ligands and the biomolecules and DNA backbone electrostatic interactions. Whilst salt bridges are known to contribute to the stability of proteins,[58] the coordination of charged residues with ions and ligands is particularly important when studying the binding specificities of small molecules. Lastly, since this work enables the inclusion of DNA in the atomistic graph, we include backbone elctrostatic interactions, which are partly responsible for the bending stiffness of DNA.[59]

Here, salt bridges are defined as special cases of hydrogen bonds that exist when both acceptor and donor are charged and thus, we again use methodology inspired by FIRST.[14] Therefore, the same set of geometric constraints as described above for hydrogen bonds is applied, but with an additional restriction: The angles formed by hydrogen, acceptor and any atoms bonded to acceptor have to be larger than 80°. Finally, the resulting edges are then weighted through a potential similar to that of hydrogen bonds (see equation S3), which has a deeper minimum in line with the view that salt bridges are slightly stronger interactions than hydrogen bonds.

Electrostatic interactions between ions/ligands and the biomolecules are often encoded within the `LINK` entries of the PDB file, which can then be used as a straightforward edge detection step. Weighting is done via a standard Coulomb potential which implicitly models the atom centres as point charges as described by equation S4. Partial charges for standard

residues were pre-calculated with the OPLS-AA force field[48] and charges for non-standard residues as well as small-molecule ligands were obtained from the GlycoBioChem PRODRG2 Server.[60]

Additionally, we introduce another type of electrostatic interactions only applicable to DNA. Unlike amino acids, nucleotides are highly charged, with most of the net negative charge residing in the phosphates of the DNA backbone.[59] Here, we model this interaction by creating edges between the phosphates of consecutive nucleotides in DNA chains. To weight the edges, we apply a potential for DNA backbone interactions which includes the Manning counterion and Debye screening effect,[59,61–63] given by equation S5.

Hydrophobic interactions Hydrophobic interactions play a very important role in proteins as a stabilising force[47,64–67] within the structure. Due to their importance and significance for many processes surrounding biomolecules, it is essential for hydrophobic interactions to be included in the construction of the biomolecular graph. However, the computational modelling of hydrophobic interactions is poorly understood in the literature. In parts, this is due to the many-body effect,[47,65] which is an intrinsic property of hydrophobic interactions. It is being suggested that hydrophobic interactions result not only from local interactions but are due to global effects that go beyond isolated pairs of atoms. Commonly, hydrophobic interactions are defined to be the association of non-polar solutes in aqueous solution.[68,69]

In the following, we propose a method of deterministically modelling hydrophobic interactions whilst taking both their global as well as their local aspects into consideration. To this end, we first select a pool of candidate interactions using a set of constraints. Each possible hydrophobic interaction is then assigned a weight. Finally, we sparsify this weighted subgraph using the Relaxed Minimum Spanning Tree (RMST) method proposed by Beguerisse-Diaz et al.[33] which has been previously shown to successfully describe both the local and global information of networks,[70] thus allowing us to capture the properties of the multiscale

hydrophobic effect.

Prior to calculating edge weights for hydrophobic interactions, a preliminary set of candidate interactions is determined through a set of constraints.[14] As with hydrogen bonds, this applies a preliminary filter, ensuring that only physically realistic pairs of atoms are considered to form a hydrophobic interaction: (i) Only carbon and sulfur atoms can be part of a hydrophobic interaction. (ii) Each atom's (covalent) neighbours can only be either carbon, sulfur or hydrogen. This is to ensure that no hydrophilic moieties (which usually contain oxygen or nitrogen) are in the immediate vicinity. (iii) As we are mainly interested in interactions between different residues, we do not allow hydrophobic interactions within amino acids, i.e. both atoms in a hydrophobic interaction have to be in different residues. (iv) Both atoms must not be in each other's third neighbourhood, i.e. the shortest path between them along the network of covalent bonds must be of length > 3.

In order to compute a weight for each possible hydrophobic interaction, we adapt and apply a potential of mean force for hydrophobic interactions (derived and fitted experimentally by Lin et al.[47]) to calculate a corresponding weight for the interaction. In their work, the authors describe a potential, which is the sum of three Gaussian functions, reproduced in equation S6. Figure S.F2 shows the potential, including its four constituent summands. Note that this potential has been successfully used in our previous graph construction approach to parametrise the hydrophobic edges.[26,28,30] Here, we have improved on the determination of hydrophobic edges as will be explained in more detail below.

We now compute the Relaxed Minimum Spanning Tree[33] of the set of all candidate hydrophobic interactions weighted by the potential. This algorithm was originally proposed as a method for sparsifying a fully connected set of similarity measures and considers both local and global aspects of the data. In the context of hydrophobic interactions, the goal is to reduce the complexity of the interactional system. In neighbourhoods dominated by stronger interactions, comparably weak interactions have little contribution and are therefore filtered out. On the other hand, such weak interactions are retained in regions of lower

strength interactions (typically long-range interactions), so as to prevent the loss of valuable information on long-range effects. Particularly the last point sets this technique apart from a simplistic energetic cutoff. As a consequence of the reduction in complexity, further computations are much more feasible.

The algorithmic structure of RMST is best visualised and explained through pseudo-code, which is given in algorithm 1 below. A few lines require further highlighting. In line 6, we construct a minimum spanning tree (MST) making use of Prim's algorithm.[71] As the MST minimises the sum of the edge weights, we obtain a subset of hydrophobic interactions which is both fully connected and optimised to contain the strongest (most negative) interactions, representing the globally influential subset of hydrophobic interactions. Both mlink_{ij} defined in line 10 and d_i defined in line 11 can be understood as measures of the local environment around a given pair of atoms. On top of the MST, the algorithm allows certain edges that fulfil the criterion set in line 12, i.e. edges that are significantly "strong" compared to their neighbourhood. This can be understood as "relaxing" the tree. Finally, γ represents a free parameter, which can be pruned to include more (larger γ) or less (smaller γ) hydrophobic interactions. In the context of hydrophobic interactions, $\gamma = 0.1$ typically, as this achieves a total energetic contribution of all hydrophobic interactions that is comparable to that of all hydrogen bonds. This relationship has been observed in the literature.[55,72]

π-π stacking interactions The existence of π-π stacking interactions, or simply π-π interactions, has been well established[49] and they are known to greatly contribute to the stability and structure of nucleic acids, hence influencing their function.[73] π-π interactions are defined to be the total interaction between two aromatic moieties, resulting from the interactions between π-electrons.[73]

The accurate description of this type of interaction is crucial to the graph construction for either isolated DNA or protein-DNA complexes and was developed by Delmotte et al.[29,61] It uses the well-established potential proposed by Hunter and Sanders[49] for the Van der Waals

Algorithm 1 Relaxed Minimum Spanning Tree for the selection of hydrophobic interactions

Require: $(V, E) \leftarrow$ set of nodes and edges in graph of possible hydrophobic interactions

function $w(i, j)$
    return weight of edge $(i, j) \leftarrow w_{\text{hydrophobic}}(i, j)$
end function ▷ Note that $w(i, j) < 0 \; \forall i, j$.

procedure RMST(V, E, γ)
    From (V, E), obtain a minimum spanning tree (V_{MST}, E_{MST})
    $(V_{RMST}, E_{RMST}) \leftarrow (V_{MST}, E_{MST})$
    for each i, j such that $(i, j) \in E$ and $(i, j) \notin E_{MST}$ **do**
        Obtain path $\{(i, k), (k, l), ..., (m, j)\}$ in E_{MST}
        $\text{mlink}_{ij} \leftarrow \max\{w(i, k), w(k, l), ..., w(m, j)\}$
        $d_i \leftarrow \min_k\{w(i, k)\}$ and $d_j \leftarrow \min_k\{w(j, k)\}$
        if $w(i, j) < \text{mlink}_{ij} + \gamma|d_i + d_j|$ **then**
            $E_{RMST} \leftarrow E_{RMST} \cup (i, j)$
        end if
    end for
    return (V_{RMST}, E_{RMST})
end procedure

contribution and the electrostatic contribution term from Warshel et al. [50] as can be seen in equation S7. This potential is parameterised using values obtained by the authors of Ref. [49]

Unlike the other weighting functions in this work, we consider pairs of nucleobases here, i.e. we calculate the total interaction energy between every pair of atoms each belonging to one of two nucleobases. This is computed for each possible pair of nucleobases, regardless of proximity. This ensures that the correct energies are calculated for standard as well as non-canonical DNA structures such as G-quadruplexes, where interacting nucleobases are not necessarily consecutive. The resulting energy is then subject to a threshold calculated from the thermal energy at room temperature (300K) corresponding to approx. -0.596kcal/mol. [74] In standard duplex DNA, this makes it highly unlikely that non-consecutive nucleobases are considered. Finally, the total energy of the interaction is spread across the moieties, rather than creating one overpowering edge and is distributed equally, for simplicity, onto the six edges connecting each atom of one nucleobase's benzene ring to its equivalent on the other nucleobase.

Uniquification and output

Once each bond/interaction detection routine has successfully been completed, the edges of the graph are "uniquified". If two different types of bonds or interactions between the same two atoms were detected, the algorithm will merge the duplicate edges and add the energies up, implicitly assuming additivity between bond types.

Following this, the resulting graph is output in a number of formats. Primarily, `BagPype` produces two types of output: (i) two spreadsheets detailing all nodes and edges respectively (in `.csv` file format) and (ii) a `Graph` object based on `networkx`[75] (a popular Python package for the study of complex networks). Whilst the former is very general-purpose and can be read by a large variety of software for further analysis, such as computing statistical features, the latter is particularly well-suited for graph-theoretic analysis, due to the large range of output options (e.g. in Graph Exchange XML Format or GraphML), implemented in `networkx` which facilitate direct interfacing with other network science software.

Results & Discussion

The results presented in this work are two-fold: First, we show the computational feasibility of the graph construction as implemented in `BagPype` by obtaining timing data for a large sample of proteins, nucleic acids (NAs) and protein-NA complex structures across the PDB. As mentioned above, our implementation is single-threaded and does not make use of any parallel computing packages. On multi-core processing units, it is therefore possible to run multiple graph construction instances in parallel, making our method suitable for fast high-throughput analysis. Second, we show how graph theoretic measures can be directly applied on the graph construction by drawing exemplatory methods from graph/network theory and biophysics.

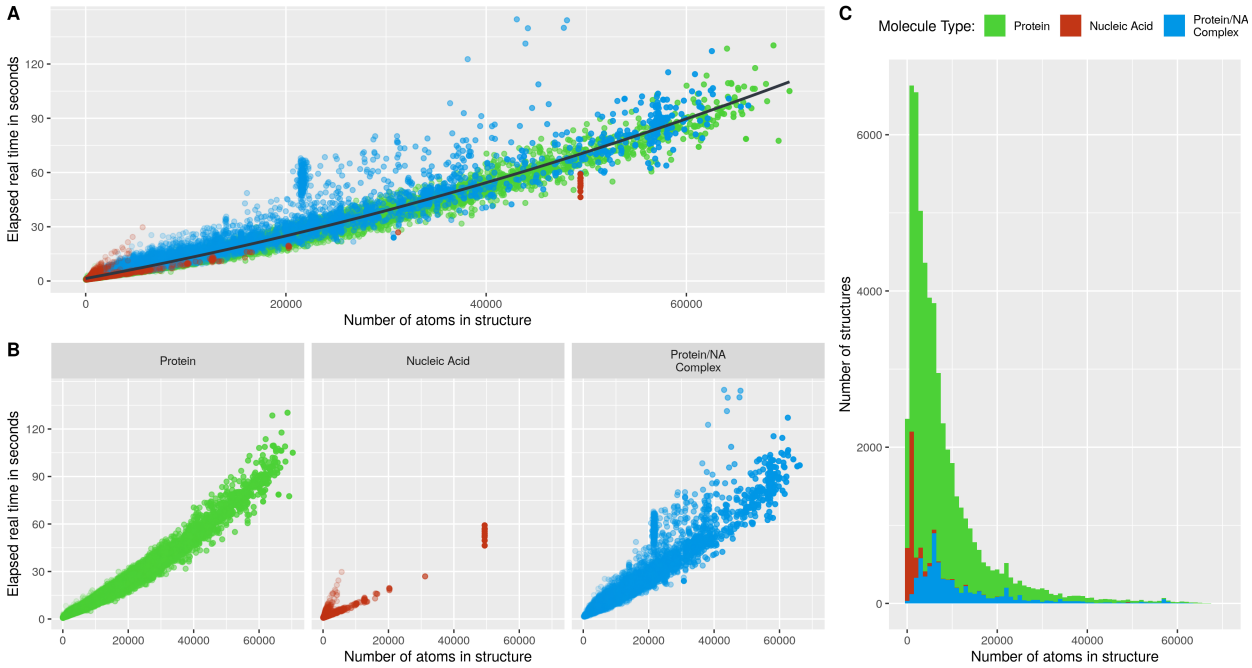

Figure 2: Scatter plots of measured wall-clock time of the graph construction against number of atoms for 54,447 PDB structures containing proteins, nucleic acids and protein-nucleic acid complexes on a standard desktop. **A:** In addition to the overall scatter plot, a linear second-order polynomial regression based on the entire dataset is shown as a solid black line (see main text for details). **B:** The graph construction time for proteins, nucleic acids and protein-nucleic acid complexes is shown separately. **C:** A histogram (binwidth: 1000) shows the size distribution (in terms of number of atoms) of each dataset used here and reflects the resolved structures available in the PDB to date.

Computational feasibility

The continuous advances in experimental techniques allow for larger biomolecular complexes to be resolved at better resolutions in complex environments. This calls for computational methods that scale well as structures become larger. In the following, we show that our graph construction algorithm `BagPype` remains computationally viable even for large protein structures without compromising on the level of detail. To that end, we provide data-based evidence that our implementation of the above described method is subject to scaling in low-order polynomial time. We obtain three separate data sets by mining the PDB (as of March 2020) according to the following specifications: (i) All structures containing proteins only with a molecular weight less than 500 kDa (approx. 70,000 atoms after adding hydrogens),

(ii) all structures containing nucleic acids only and (iii) all structures containing both protein and nucleic acids with a molecular weight less than 500kDa. To reduce redundancy in the protein-only data set, we applied a 90% threshold for sequence similarity using BLAST[76] and, when resolved by X-Ray crystallography, only consider proteins with resolution lower than 3Å. This resulted in (i) 44,536, (ii) 3,441 and (iii) 6929 structures in the respective categories, i.e. a total of 54,906 PDB files. Note that in order to facilitate a high-throughput analysis, all PDB files are stripped of non-standard residues in these datasets.

For each structure in all three data sets, we carry out the entire graph construction work-flow as described in figure 1 and measure the elapsed wall-clock time (real time). The executions were run in parallel on a workstation with the following technical specifications: 2×Intel® Xeon® CPU E5-2690 v3 @ 2.60GHz, 256GB RAM. Due to some files containing non-standard entries or other non-standard text, not all files could be processed, leading to a final number of 54,447 structures (44,274 of proteins only, 3,371 of NA only and 6,802 of protein-NA complexes) used for the analysis below.

To obtain the time complexity, we reject a first-order linear model because the implementation contains nested looping, and instead consider a second-order polynomial ansatz. This is also theoretically consistent with the general structure of the algorithm, which consists, for almost all bond types, of one loop over all atoms as well as a second loop over each atom's neighbourhood searching for possible bonding partners.

As can be seen from figure 2A, a second order polynomial fit of the form $t = \beta_0 + \beta_1 n + \beta_2 n^2$ gives a good approximation. Here, t stands for the total execution time in seconds and n denotes the total number of atoms in the biomolecule. We find that $\beta_0 = 1.388021$, $\beta_1 = 1.031767 \times 10^{-3}$ and $\beta_2 = 7.330826 \times 10^{-9}$, with coefficient of determination $R^2 = 0.9413$.

Intuitively, the intercept β_0 can be understood as the algorithm's overhead, i.e. loading additional modules, reading the file etc., which in this case takes (on average) around 1.39 seconds, while β_1 can be interpreted to be the linear scaling in seconds per number of atoms. β_2 represents the quadratic scaling factor arising from more complex parts of the algorithm.

The numerical values are of lesser importance and indeed will depend on the hardware used, but rather it is the second-order polynomial behaviour which is of significance. Note that a third-order polynomial fit as well as higher orders were rejected, as this introduced a negative coefficient for the cubic term, which is not only inappropriate for our data sets, but also indicative of overfitting.

Furthermore, since structures of approximately less than $20,000$ atoms still dominate the PDB currently (see figure 2C), the quadratic term does not play a significant role for almost all structures of the PDB, amounting to a mere few seconds (e.g. $7.33 \times 10^{-9} \times 20,000^2 \approx$ 3s with the hardware used here). Therefore, for structures of that size and smaller, our methodology achieves a near-linear scaling in time complexity.

The two distinct behaviours of nucleic acids in Figure 2B, reflect the stacked interactions implemented for DNA, which are currently absent in RNA but will be included in a future release of the software.

Furthermore, it is worth noting the difference between structures containing proteins only vs protein-NA complexes. The latter show running times far more varied than the former, which is of course due to the nucleic acids adding to the execution time, especially for larger structures. Because of the added types of interactions for DNA (π-π stacking and backbone interactions), structures involving DNA take longer to process.

Finally, in addition to the the timing complexity trends we see that for most structures, the atomistic graph can be obtained from a few seconds to a few minutes which makes the methodology highly suitable for large scale as well as high-throughput analyses.

Case study: Multiscale features of the *lac* operon repressor protein-DNA complex

In this section, we aim to recover the macro-molecular structural buildup from only the atomistic graph constructed through the software described in this paper. To this end, the Markov Stability framework,[27,77,78] a highly versatile, multi-scale graph clustering method-

ology, has been successfully applied in the past, to extract various properties of atomistic protein graphs, ranging from uncovering levels of organisation within proteins[30] to finding highly disruptive mutations.[28,31,79] An inherent strength of this methodology compared to similar graph clustering algorithms is its independence from particular scales. Through a parameter, called the *Markov time*, we are able to "zoom" through multiple resolutions, finding clusters (also known as partitions or communities and used synonymously throughout the text here) ranging from small to large.

Whilst the full extended details of the method can be left to more technical papers, we necessarily need to introduce two measures with respect to Markov time, by which a "good" clustering can be identified: (i) The number of communities (NC): At each Markov time t, we calculate the optimal clustering at the corresponding resolution. Here, we are looking for "plateaus", which indicate that a particular x-way partition is persistent across a range of Markov time. (ii) The variation of information (VI): Since the greedy optimisation algorithm is run multiple times at each Markov time, the optimal solution may not be the only solution. Thus, in order to quantify the spread, we calculate the VI, a measure of distance between two clusterings originating from probability theory and take the average. Here, a "good" result is signified by a low VI value, meaning that there was good agreement between the various solutions.

In figure 3, this method can be seen in action on the *lac* repressor protein. As expected, Markov Stability is able to uncover the structural buildup at the chemical bond level at low Markov time. The next place of low VI firmly corresponds to amino acids, which is then followed by a good clustering into secondary structure elements, such as α-helices. Finally, at large Markov time, a particularly exquisit 6-way clustering emerges, which is persistent across a very long range of Markov times, accompanied by a steady decline in VI. The very last end of Markov times corresponds to a very stable 3-way clustering, splitting the protein into its two core domains as well as the DNA-binding domains together with their operator DNA. From a biological point of view, these are all sensible clusterings, taking full

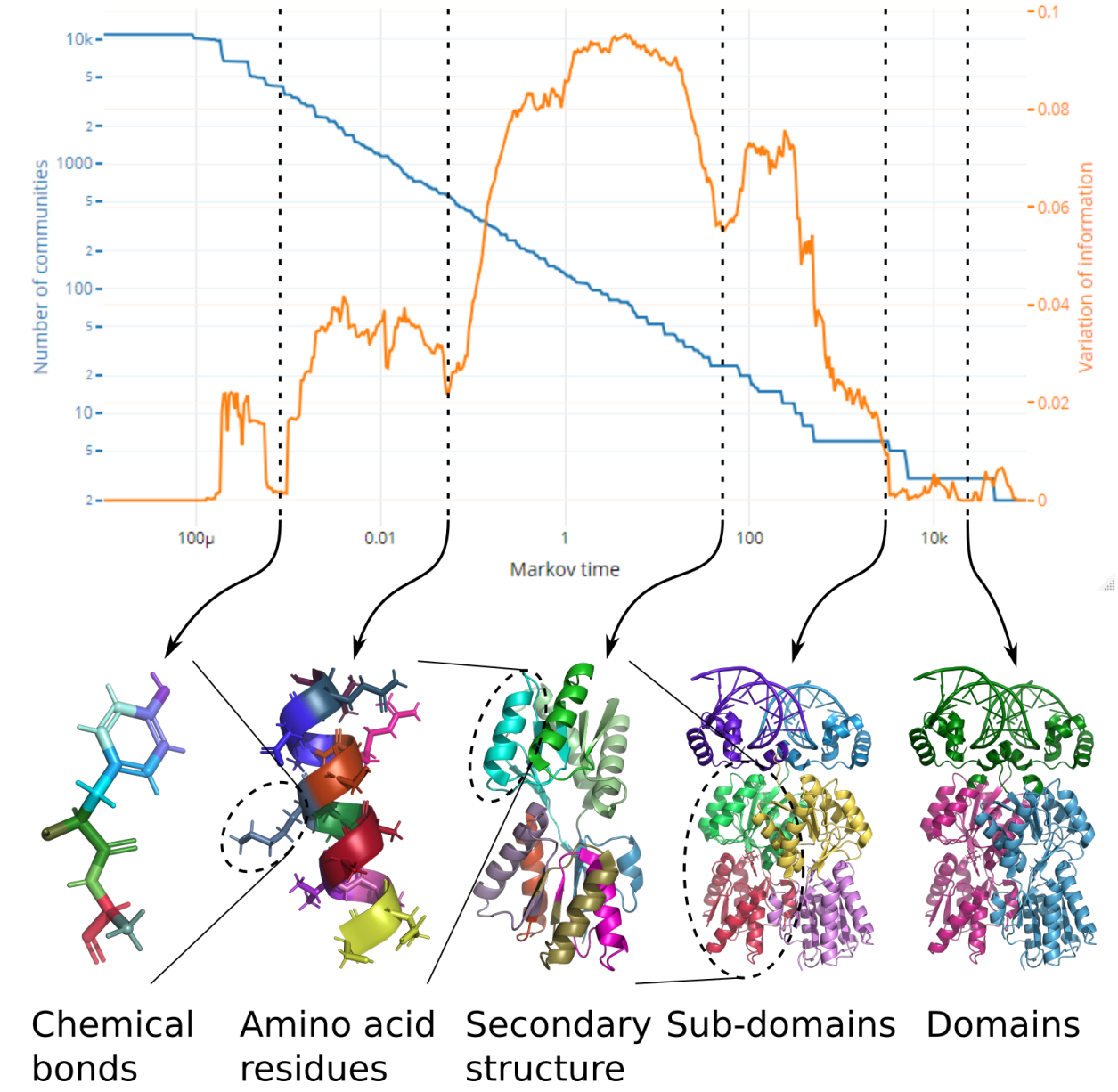

Figure 3: The *lac* repressor protein (dimer) with operator DNA seen through the Markov Stability method,[27] a very versatile graph clustering tool. The main parameter, *Markov time*, acts as a "zooming lens", exploring a wide range of community sizes of the protein. At very small *Markov times*, chemical bonds form individual communities and at very large *Markov times*, large-scale protein domains are uncovered. The results are described by two measures with respect to *Markov time*: the number of communities (NC, shown in blue) as well as variation of information (VI, shown in orange). Meaningful communities are characterised by persistent plateaus in the NC curve as well as low VI. As can be seen from the figure, a wide range of biologically relevant substructures are found.

Table 2: Table comparing border-line residues between those used in the literature and those found by Markov Stability. †: Not clearly defined.

Border between which domains	Markov Stability	Literature[80]
headpiece+hinge / NH_2-subdomain	58/59	58/59
NH_2-subdomain / CO_2-subdomain	161/162	161/162
	291/292	289/293†
	319/320	320/321

advantage of the multi-scale nature of the method. Finally, we take a closer look at the 6-way partitioning, and in particular the border residues between different clusters. We find that we are in almost perfect agreement with the commonly used split into domains taken from the literature surrounding the particular structure used here.[80] For a detailed overview, see table 2.

Conclusions

In this work, we introduced novel methodology for the construction of an energy-weighted, atomistic graph from experimentally obtained biochemical structures. Besides covalent bonds, further types of weak interactions are included in the graphs: hydrogen bonds, hydrophobic interactions, a range of electrostatic interactions as well as π-π stacking for DNA. For all types of bonds/interactions, we draw on existing methodologies and adapt them for this purpose. In particular, we proposed a novel way of including both the local and the global scales of hydrophobic interactions through the application of RMST, a recently devised graph sparsification tool. Through the addition of DNA π-π stacking interactions, we are now able to apply the graph construction not just to proteins, but DNA as well as protein-DNA complexes.

In addition to the theretical work, we also provide an open-source implementation in Python. With `BagPype`, we created a self-contained package for constructing atomistic graphs from files in the PDB format. Its output is highly versatile, allowing for a wide range of analysis and applications to build on `BagPype`.

We then demonstrated both the computational feasibility as well as showcased our methodology by applying graph-theoretical analysis tools. To this end, we have computed execution times for a vast set of biochemical structures obtained from the PDB. We found that the algorithm scales as a slow-growing second order polynomial. Even large biomolecules can be processed within minutes on a standard desktop or workmachine. Furthermore, we applied a recently developed multiscale graph clustering method, Markov Stability,[27] to a well-studied protein, the *lac* operon repressor. We found that the method was able to fully recover multiple levels of structural composition, ranging from individual amino acid residues to sub-domains and domains.

We therefore showed that our methodology is not only fast and efficient but also remains true to reality. Therefore, we are able to graph-theoretically encode atomistic physicochemical properties of biochemical molecules in a fast manner without compromising on accuracy or level of detail.

With this work, we hope to provide the foundational basis for the application of established ideas from graph theory as well as network science to the detailed graphs of biomolecular structures as described here, as well as encouraging the further development of analysis methodology, that is purpose-built for the investigation of open questions relevant to biochemistry. Since our methodology is implemented in a general-purpose manner, it can be combined with further approaches and techniques. Graph theory and network science provide a vast arsenal of useful tools, many of which have been proven to be applicable to various scopes outside mathematics. Much remains unexplored with regards to atomistic graphs, and so we anticipate that "atomistic, biochemical graph theory" will be of increasing importance in the future.

Acknowledgement

F.S. was supported by a PhD studentship of the EPSRC Centre for Doctoral Training (grant

EP/L015498/1) at the Institute of Chemical Biology, Department of Chemistry, Imperial College London. We also acknowledge the Engineering and Physical Sciences Research Council (EPSRC) award EP/N014529/1 supporting the EPSRC Centre for Mathematics of Precision Healthcare.

Supporting Information Available

Functional forms of edge weighting potentials

In its simplest description, a graph G can be denoted as $\big(V, E\big)$, a collection of two sets describing the *nodes* (also known as *vertices*) and *edges* contained within the graph.[81,82] Two nodes i, j are connected if and only if $(i, j) \in E$. Furthermore, in a *weighted graph*, there exists a function $w\colon E \mapsto \mathbb{R}$, which maps each edge to a real number (the *edge-weight*).[83] In this section, we give the explicit weighting equations associated with each bond/interaction type.

Covalent bonds

Table S.T1: Bond dissociation energies in kcal/mol used to weight covalent bonds in the graph construction. Values (excerpt) taken from Ref.[43]

Bond	Energy (kcal/mol)	Bond	Energy (kcal/mol)
H—C	98.3	C—O	85.5
H—N	92	C=O	190.9
H—O	109.6	C—P	63
H—P	77	C—S	65
H—S	87	P—O	80
C—C	82.6	P=O	130
C=C	144	S—O	127.2
C—N	72.8	N=O	145
C=N	147	N—O	48

Making use of notation introduced at the outset, we can write:

$$w_{\text{covalent}}(i,j) \coloneqq \begin{cases} 98.3, & \text{if } i \text{ is hydrogen and } j \text{ is carbon and vice versa} \\ 92, & \text{if } i \text{ is hydrogen and } j \text{ is nitrogen and vice versa} \\ \dots & \text{see table S.T1} \end{cases} \tag{S1}$$

Hydrogen bonds

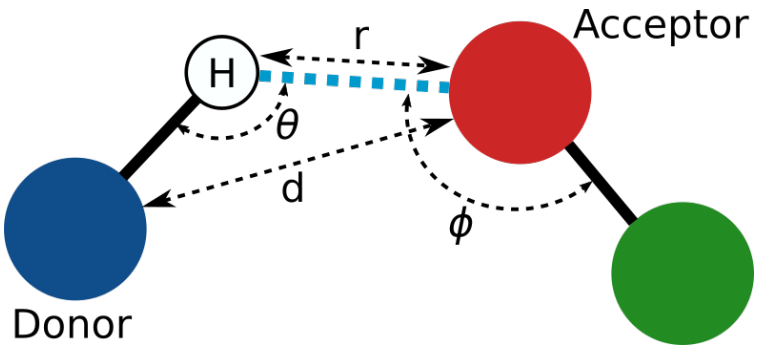

Figure S.F1: Hydrogen bonds and variables used in the potential. The diagram shows a donor atom (in blue), a hydrogen bonded covalently to the donor (in white), an acceptor atom (in red) and a neighbouring atom to the acceptor (in green). The resulting hydrogen bond is shown as a dotted lightblue line. Each solid line represents a covalent bond. Lastly, dashed lines represent variables as well as angles used in the hydrogen bond potential.

$$w_{\text{hydrogen}}(i,j) \coloneqq V_0 \left\{ 5\left(\frac{d_0}{d_{ij}}\right)^{12} - 6\left(\frac{d_0}{d_{ij}}\right)^{10} \right\} F(\theta,\phi,\gamma), \tag{S2}$$

where $V_0 = 8$ kcal/mol, $d_0 = 2.8$Å and d_{ij} is the distance between donor and acceptor (denoted by i,j) as shown in Figure S.F1. The functional form of the angular term $F(\theta,\phi,\gamma)$ depends on the hybridisation of both donor and acceptor as shown in table S.T2. Note that d_{ij} (d in the figure), r, θ and ϕ are as shown in Figure S.F1.

Table S.T2: The functional form of the angular term in the hydrogen bond potential depending on the hybridisation of donor and acceptor. θ and ϕ are angles as shown in Figure S.F1 and γ is the maximum angle between the normals of the planes defined by the covalent bonds of the donor and base atoms (coloured green in Figure S.F1). In cases where there is more than one atom bonded to the acceptor, the maximum ϕ is used. Note that all angles are in radians.

Donor	**Acceptor**	**Functional form**
sp^3	sp^3	$F = e^{-(\pi-\theta)^6} \cos^2\theta \cos^2(\phi - \frac{109.5}{180}\pi)$
sp^3	sp^2	$F = e^{-(\pi-\theta)^6} \cos^2\theta \cos^2\phi$
sp^2	sp^3	$F = e^{-2(\pi-\theta)^6} \cos^4\theta$
sp^2	sp^2	$F = e^{-(\pi-\theta)^6} \cos^2\theta \cos^2(\max(\phi,\gamma))$

Salt bridges

$$w_{\text{salt}}(i,j) \coloneqq V_S\left\{5\Big(\frac{d_S}{d_{ij}+x}\Big)^{12} - 6\Big(\frac{d_S}{d_{ij}+x}\Big)^{10}\right\}, \tag{S3}$$

where $V_S = 10$ kcal/mol, $d_S = 3.2$Å, d_{ij} is as above and $x = 0.375$Å.

Electrostatic interaction between ions/ligands and the biomolecules

$$w_{\text{electrostatic}}(i,j) \coloneqq \frac{332}{\epsilon}\frac{q_i q_j}{d_{ij}}, \tag{S4}$$

where $\epsilon = 4$.[84] q_i and q_j represent the point charges of the two atoms (i,j) that are to be linked. As before, d_{ij} denotes the distance between the two atoms.

DNA backbone electrostatic interactions

$$w_{\text{backbone}}(i,j) \coloneqq \frac{332\delta^2}{\epsilon}\frac{e^{-\kappa d_{ij}}}{d_{ij}}, \tag{S5}$$

where the net effective charge $\delta = 0.24$, the Debye screening parameter $\kappa = 0.329\sqrt{c}$ and $c = 0.1$mol/l for monovalent salts such as NaCl[59] and $\epsilon = 4$.

Hydrophobic interactions

$$w_{\text{hydrophobic}}(i,j) \coloneqq \sum_{k=1}^{3} h_k \exp\Big(-\Big(\frac{d_{ij}-c_k}{w_k}\Big)^2\Big), \tag{S6}$$

where i and j denote a candidate pair of atoms and k denotes the index of the three Gaussians as defined in table S.T3. d_{ij} indicates the spatial distance between i and j. The parameters for the Gaussian functions are taken from Ref.[47] (see table S.T3).

π-π interactions

$$w_{\text{stacking}}(b_1,b_2) \coloneqq \sum_{i\in b_1,\, j\in b_2}\Bigg[\underbrace{K_iK_j\Big[C\exp\big(-\alpha z\big) - \frac{A}{z^6}\Big]}_{\text{Van der Waals contribution}} + \underbrace{\sum_{kl}\Big[\frac{332}{\epsilon}\frac{q_i^k q_j^l}{d_{ij}^{kl}}\Big]}_{\text{Electrostatic contribution}}\Bigg], \tag{S7}$$

Table S.T3: Parameters used in the hydrophobic potential. Taken from Ref.[47] The parameters for the Gaussian functions are c_k, w_k and h_k which denote the centre (position of the peak), width (the standard deviation) and height (height of the peak) respectively.

Gaussian function parameters

	1st Gaussian $k=1$	**2nd Gaussian** $k=2$	**3rd Gaussian** $k=3$
c_k	3.81679	5.46692	7.11677
w_k	1.68589	1.39064	1.57417
h_k	-0.73080	0.20016	-0.09055

where

$$z = \frac{d_{ij}}{d_{ij}^W} \quad \text{and} \quad d_{ij}^W = \sqrt{\left(2d_i^W\right)\left(2d_j^W\right)}$$

The distance between the two atoms is denoted by d_{ij}. Furthermore, k and l are parameters, each corresponding to the three point charges associated with each atom: a σ point charge at the centre of the atom and two π point charges at 0.47Å above and below the aromatic plane. In the above equation, k stands for the three point charges of atom i, and l stands for the three point charges of atom j,

d_{ij}^{kl} is the distance between each pair of point charges. The remaining constants (all taken from Caillet and Claverie,[85] except ϵ which was determined in Gilson and Honig[84]) are as follows:

$$\alpha = 12.35, \quad A = 0.214, \quad C = 47 \times 10^3, \quad \epsilon = 4$$

r_{X}^W denotes the van der Waals radius of atom X:

$$r_{\mathrm{H}}^W = 1.2\text{Å}, \quad r_{\mathrm{C(aliphatic)}}^W = 1.7\text{Å}, \quad r_{\mathrm{C(aromatic)}}^W = 1.77\text{Å},$$
$$r_{\mathrm{N}}^W = 1.6\text{Å}, \quad r_{\mathrm{O}}^W = 1.5\text{Å}$$

The parameters K_{X} depend on the element of the atom X and denote the energy minimum

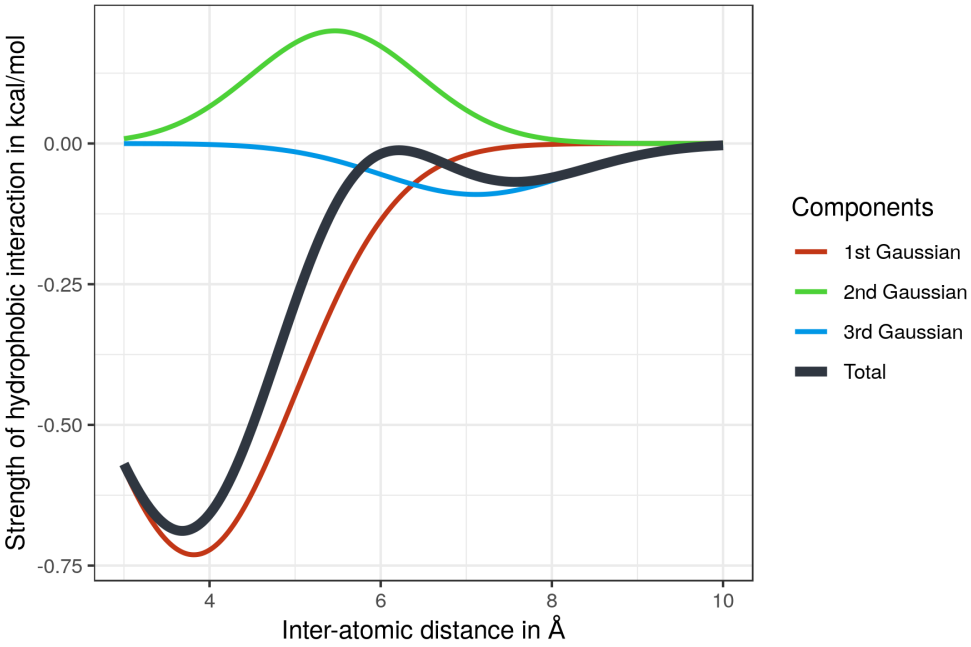

Figure S.F2: The hydrophobic potential used in our graph construction is shown together with its respective summand functions. The overall potential is shown by a thick black line, the 1st, 2nd and 3rd Gaussian functions are shown in red, green and blue respectively.

according to the atomic species:[85]

$$K_{\mathrm{H}} = 1, \quad K_{\mathrm{C}} = 1, \quad K_{\mathrm{N}} = 1.18, \quad K_{\mathrm{O}} = 1.36$$

Depending on the base (one of adenine, guanine, cytosine or thymine*) and whether the point charge is σ or π, q_i^k can take a different value.

*Note that only standard DNA is currently considered for π-π stacking.

References

(1) Levitt, M.; Warshel, A. Computer simulation of protein folding. *Nature* **1975**, *253*, 694–698.

(2) Ayton, G. S.; Noid, W. G.; Voth, G. A. Multiscale modeling of biomolecular systems: in serial and in parallel. *Current Opinion in Structural Biology* **2007**, *17*, 192–198.

(3) Kmiecik, S.; Gront, D.; Kolinski, M.; Wieteska, L.; Dawid, A. E.; Kolinski, A. Coarse-Grained Protein Models and Their Applications. *Chemical Reviews* **2016**, *116*, 7898–7936.

(4) Wohlert, J.; Berglund, L. A. A Coarse-Grained Model for Molecular Dynamics Simulations of Native Cellulose. *Journal of Chemical Theory and Computation* **2011**, *7*, 753–760.

(5) Piovesan, D.; Minervini, G.; Tosatto, S. C. The RING 2.0 web server for high quality residue interaction networks. *Nucleic Acids Research* **2016**, *44*, W367–W374.

(6) Ryslik, G. A.; Cheng, Y.; Cheung, K.-H.; Modis, Y.; Zhao, H. A graph theoretic approach to utilizing protein structure to identify non-random somatic mutations. *BMC Bioinformatics* **2014**, *15*, 86.

(7) Di Paola, L.; Giuliani, A. Protein contact network topology: a natural language for allostery. *Current Opinion in Structural Biology* **2015**, *31*, 43–48.

(8) Bahar, I.; Atilgan, A. R.; Erman, B. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design* **1997**, *2*, 173–181.

(9) Atilgan, A.; Durell, S.; Jernigan, R.; Demirel, M.; Keskin, O.; Bahar, I. Anisotropy of Fluctuation Dynamics of Proteins with an Elastic Network Model. *Biophysical Journal* **2001**, *80*, 505–515.

(10) López-Blanco, J. R.; Chacón, P. New generation of elastic network models. *Current Opinion in Structural Biology* **2016**, *37*, 46–53.

(11) Chodera, J. D.; Noé, F. Markov state models of biomolecular conformational dynamics. *Current Opinion in Structural Biology* **2014**, *25*, 135–144.

(12) Wales, D. J. Calculating rate constants and committor probabilities for transition networks by graph transformation. *The Journal of Chemical Physics* **2009**, *130*, 204111.

(13) Tirion, M. M. Large Amplitude Elastic Motions in Proteins from a Single-Parameter, Atomic Analysis. *Physical Review Letters* **1996**, *77*, 1905–1908.

(14) Jacobs, D. J.; Rader, A.; Kuhn, L. A.; Thorpe, M. Protein flexibility predictions using graph theory. *Proteins: Structure, Function, and Genetics* **2001**, *44*, 150–165.

(15) Thorpe, M.; Lei, M.; Rader, A.; Jacobs, D. J.; Kuhn, L. A. Protein flexibility and dynamics using constraint theory. *Journal of Molecular Graphics and Modelling* **2001**, *19*, 60–69.

(16) Santana, C. A.; Silveira, S. d. A.; Moraes, J. P. A.; Izidoro, S. C.; de Melo-Minardi, R. C.; Ribeiro, A. J. M.; Tyzack, J. D.; Borkakoti, N.; Thornton, J. M. GRaSP: a graph-based residue neighborhood strategy to predict binding sites. *Bioinformatics* **2020**, *36*, i726–i734.

(17) García-Domenech, R.; Gálvez, J.; de Julián-Ortiz, J. V.; Pogliani, L. Some New Trends in Chemical Graph Theory. *Chemical Reviews* **2008**, *108*, 1127–1169.

(18) Basak, S. C.; Natarajan, R.; Sinha, D. K. *Applied Mathematics*; Springer, New Delhi, 2015; Chapter 12, pp 141–148.

(19) Grant, W. P.; Ahnert, S. E. Modular decomposition of protein structure using community detection. *Journal of Complex Networks* **2019**, *7*, 101–113.

(20) Bramer, D.; Wei, G.-W. Multiscale weighted colored graphs for protein flexibility and rigidity analysis. *The Journal of Chemical Physics* **2018**, *148*, 054103.

(21) Zacharias, M. Combining coarse-grained nonbonded and atomistic bonded interactions for protein modeling. *Proteins: Structure, Function, and Bioinformatics* **2013**, *81*, 81–92.

(22) Neri, M.; Anselmi, C.; Cascella, M.; Maritan, A.; Carloni, P. Coarse-Grained Model of Proteins Incorporating Atomistic Detail of the Active Site. *Physical Review Letters* **2005**, *95*, 218102.

(23) Fassio, A. V.; Santos, L. H.; Silveira, S. A.; Ferreira, R. S.; de Melo-Minardi, R. C. nAPOLI: a graph-based strategy to detect and visualize conserved protein-ligand interactions in large-scale. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **2019**, *17*, 1–1.

(24) Estrada, E.; Bonchev, D. *Handbook of Graph Theory*; 2013; pp 1538–1558.

(25) Sarkhel, S.; Desiraju, G. R. N—H. . . O, O—H. . . O, and C—H. . . O hydrogen bonds in protein-ligand complexes: Strong and weak interactions in molecular recognition. *Proteins: Structure, Function, and Bioinformatics* **2003**, *54*, 247–259.

(26) Amor, B. R.; Schaub, M. T.; Yaliraki, S. N.; Barahona, M. Prediction of allosteric sites and mediating interactions through bond-to-bond propensities. *Nature Communications* **2016**, *7*, 12477.

(27) Delvenne, J. C.; Yaliraki, S. N.; Barahona, M. Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences* **2010**, *107*, 12755–12760.

(28) Amor, B.; Yaliraki, S. N.; Woscholski, R.; Barahona, M. Uncovering allosteric pathways in caspase-1 using Markov transient analysis and multiscale community detection. *Mol. BioSyst.* **2014**, *10*, 2247–2258.

(29) Delmotte, A.; Reynolds, M.; Vilar, R.; Barahona, M.; Yaliraki, S. N. Multiscale signatures of DNA-quadruplex unfolding through atomistic graph partitioning. *Preprint* **2017**,

(30) Delmotte, A.; Tate, E. W.; Yaliraki, S. N.; Barahona, M. Protein multi-scale organization through graph partitioning and robustness analysis: application to the myosin–myosin light chain interaction. *Physical Biology* **2011**, *8*, 055010.

(31) Zhang, H.; Salazar, J. D.; Yaliraki, S. N. Proteins across scales through graph partitioning: application to the major peanut allergen Ara h 1. *Journal of Complex Networks* **2017**, *00*, 1–14.

(32) Jacobs, D. J.; Kuhn, L. A.; Thorpe, M. F. In *Rigidity Theory and Applications*; Thorpe, M. F., Duxbury, P. M., Eds.; Kluwer Academic Publishers: Boston, 2005; pp 357–384.

(33) Beguerisse-Diaz, M.; Vangelov, B.; Barahona, M. Finding role communities in directed networks using Role-Based Similarity, Markov Stability and the Relaxed Minimum Spanning Tree. 2013 IEEE Global Conference on Signal and Information Processing. 2013; pp 937–940.

(34) The Python Foundation. Available at `python.org`.

(35) Luscombe, N. M.; Austin, S. E.; Berman, H. M.; Thornton, J. M. An overview of the structures of protein-DNA complexes. *Genome Biology* **2000**, *1*, 1–1.

(36) Joyce, A. P.; Zhang, C.; Bradley, P.; Havranek, J. J. Structure-based modeling of protein: DNA specificity. *Briefings in Functional Genomics* **2015**, *14*, 39–49.

(37) Pabo, C. O.; Sauer, R. T. Protein-DNA Recognition. *Annual Review of Biochemistry* **1984**, *53*, 293–321.

(38) Alibés, A.; Nadra, A. D.; De Masi, F.; Bulyk, M. L.; Serrano, L.; Stricher, F. Using protein design algorithms to understand the molecular basis of disease caused by protein–DNA interactions: the Pax6 example. *Nucleic Acids Research* **2010**, *38*, 7422–7431.

(39) Lu, S.; Li, S.; Zhang, J. Harnessing Allostery: A Novel Approach to Drug Discovery. *Medicinal Research Reviews* **2014**, *34*, 1242–1285.

(40) Westbrook, J. D.; Shao, C.; Feng, Z.; Zhuravleva, M.; Velankar, S.; Young, J. The chemical component dictionary: complete descriptions of constituent molecules in experimentally determined 3D macromolecules in the Protein Data Bank. *Bioinformatics* **2015**, *31*, 1274–1278.

(41) Pyykkö, P.; Atsumi, M. Molecular Single-Bond Covalent Radii for Elements 1-118. *Chemistry - A European Journal* **2009**, *15*, 186–197.

(42) Pyykkö, P.; Atsumi, M. Molecular double-bond covalent radii for elements Li-E112. *Chemistry - A European Journal* **2009**, *15*, 12770–12779.

(43) Huheey, J. E.; Keiter, E. A.; Keiter, R. L. *Inorganic Chemistry: Principles of Structure and Reactivity*, 4th ed.; HarperCollins College Publishers, 1993.

(44) Dahiyat, B. I.; Benjamin Gordon, D.; Mayo, S. L. Automated design of the surface positions of protein helices. *Protein Science* **1997**, *6*, 1333–1337.

(45) Mayo, S. L.; Olafson, B. D.; Goddard, W. A. DREIDING: a generic force field for molecular simulations. *The Journal of Physical Chemistry* **1990**, *94*, 8897–8909.

(46) Beguerisse-Diaz, M.; Garduno-Hernandez, G.; Vangelov, B.; Yaliraki, S. N.; Barahona, M. Interest communities and flow roles in directed networks: the Twitter network of the UK riots. *Journal of The Royal Society Interface* **2014**, *11*, 20140940–20140940.

(47) Lin, M. S.; Fawzi, N. L.; Head-Gordon, T. Hydrophobic Potential of Mean Force as a Solvation Function for Protein Structure Prediction. *Structure* **2007**, *15*, 727–740.

(48) Jorgensen, W. L.; Tirado-Rives, J. The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *Journal of the American Chemical Society* **1988**, *110*, 1657–1666.

(49) Hunter, C. A.; Sanders, J. K. M. The nature of .pi.-.pi. interactions. *Journal of the American Chemical Society* **1990**, *112*, 5525–5534.

(50) Warshel, A.; Sharma, P. K.; Kato, M.; Parson, W. W. Modeling electrostatic effects in proteins. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* **2006**, *1764*, 1647–1676.

(51) Berman, H.; Henrick, K.; Nakamura, H. Announcing the worldwide Protein Data Bank. *Nature Structural & Molecular Biology* **2003**, *10*, 980–980.

(52) Copyright Michael Palmer, University of Waterloo.

(53) Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C. Asparagine and glutamine: Using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of Molecular Biology* **1999**, *285*, 1735–1747.

(54) Paton, K. An algorithm for finding a fundamental set of cycles of a graph. *Communications of the ACM* **1969**, *12*, 514–518.

(55) Pace, C. N.; Fu, H.; Fryar, K. L.; Landua, J.; Trevino, S. R.; Schell, D.; Thurlkill, R. L.; Imura, S.; Scholtz, J. M.; Gajiwala, K.; Sevcik, J.; Urbanikova, L.; Myers, J. K.; Takano, K.; Hebert, E. J.; Shirley, B. A.; Grimsley, G. R. Contribution of hydrogen bonds to protein stability. *Protein Science* **2014**, *23*, 652–661.

(56) Hubbard, R. E.; Kamran Haider, M. *Encyclopedia of Life Sciences*; John Wiley & Sons, Ltd: Chichester, UK, 2010.

(57) Vogt, G.; Woell, S.; Argos, P. Protein thermal stability, hydrogen bonds, and ion pairs. *Journal of Molecular Biology* **1997**, *269*, 631–643.

(58) Sarakatsannis, J. N.; Duan, Y. Statistical characterization of salt bridges in proteins. *Proteins: Structure, Function, and Bioinformatics* **2005**, *60*, 732–739.

(59) Swigon, D. In *Mathematics of DNA Structure, Function and Interactions*; Benham, C. J., Harvey, S., Olson, W. K., Sumners, D. W., Swigon, D., Eds.; The IMA Volumes in Mathematics and its Applications; Springer New York: New York, NY, 2009; Vol. 150.

(60) Schüttelkopf, A. W.; van Aalten, D. M. F. PRODRG : a tool for high-throughput crystallography of protein–ligand complexes. *Acta Crystallographica Section D Biological Crystallography* **2004**, *60*, 1355–1363.

(61) Delmotte, A. All-scale structural analysis of biomolecules through dynamical graph partitioning. Ph.D. thesis, Imperial College London, 2014.

(62) Ravishanker, G.; Auffinger, P.; Langley, D. R.; Jayaram, B.; Young, M. A.; Beveridge, D. L. *Reviews in Computational Chemistry, Volume 11*; John Wiley & Sons, Ltd, 2007; pp 317–372.

(63) Manning, G. S. The molecular theory of polyelectrolyte solutions with applications to the electrostatic properties of polynucleotides. *Quarterly Reviews of Biophysics* **1978**, *11*, 179–246.

(64) Raschke, T. M.; Tsai, J.; Levitt, M. Quantification of the hydrophobic interaction by simulations of the aggregation of small hydrophobic solutes in water. *Proceedings of the National Academy of Sciences* **2001**, *98*, 5965–5969.

(65) Shimizu, S.; Chan, H. S. Anti-cooperativity and cooperativity in hydrophobic interactions: Three-body free energy landscapes and comparison with implicit-solvent potential functions for proteins. *Proteins: Structure, Function, and Genetics* **2002**, *48*, 15–30.

(66) Erzan, A.; Tüzel, E. Hydrophobic models of protein folding and the thermodynamics of chain-boundary interactions. *Brazilian Journal of Physics* **2003**, *33*, 573–588.

(67) Baldwin, R. L. Dynamic hydration shell restores Kauzmann's 1959 explanation of how the hydrophobic factor drives protein folding. *Proceedings of the National Academy of Sciences* **2014**, *111*, 13052–13056.

(68) Ghose, A. K.; Crippen, G. M. Atomic physicochemical parameters for three-dimensional-structure-directed quantitative structure-activity relationships. 2. Modeling dispersive and hydrophobic interactions. *Journal of Chemical Information and Modeling* **1987**, *27*, 21–35.

(69) Wood, R. H.; Thompson, P. T. Differences between pair and bulk hydrophobic interactions. *Proceedings of the National Academy of Sciences* **1990**, *87*, 946–949.

(70) Liu, Z.; Barahona, M. Graph-based data clustering via multiscale community detection. *Applied Network Science* **2020**, *5*, 3.

(71) Prim, R. C. Shortest Connection Networks And Some Generalizations. *Bell System Technical Journal* **1957**, *36*, 1389–1401.

(72) Spolar, R. S.; Ha, J. H.; Record, M. T. Hydrophobic effect in protein folding and other noncovalent processes involving proteins. *Proceedings of the National Academy of Sciences* **1989**, *86*, 8382–8385.

(73) Hunter, C. A. Sequence-dependent DNA structure. *BioEssays* **1996**, *18*, 157–162.

(74) McGaughey, G. B.; Gagné, M.; Rappé, A. K. π-Stacking Interactions. *Journal of Biological Chemistry* **1998**, *273*, 15458–15463.

(75) Available at `https://networkx.github.io/`,.

(76) Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; Lipman, D. J. Basic local alignment search tool. *Journal of Molecular Biology* **1990**, *215*, 403–410.

(77) Schaub, M. T.; Delvenne, J.-C.; Yaliraki, S. N.; Barahona, M. Markov Dynamics as a Zooming Lens for Multiscale Community Detection: Non Clique-Like Communities and the Field-of-View Limit. *PLoS ONE* **2012**, *7*, e32210.

(78) Lambiotte, R.; Delvenne, J.-C.; Barahona, M. Random Walks, Markov Processes and the Multiscale Modular Organization of Complex Networks. *IEEE Transactions on Network Science and Engineering* **2014**, *1*, 76–90.

(79) Peach, R. L.; Saman, D.; Yaliraki, S. N.; Klug, D. R.; Ying, L.; Willison, K. R.; Barahona, M. Unsupervised Graph-Based Learning Predicts Mutations that Alter Protein Dynamics. *bioRxiv* **2019**, 1–19.

(80) Bell, C. E.; Lewis, M. A closer view of the conformation of the Lac repressor bound to operator. *Nature Structural Biology* **2000**, *7*, 209–214.

(81) Diestel, R. *Graph theory*, 5th ed.; Springer-Verlag: Heidelberg, 2016.

(82) Tutte, W. T. On the Problem of Decomposing a Graph into n Connected Factors. *Journal of the London Mathematical Society* **1961**, *s1-36*, 221–230.

(83) Gibbons, A. *Algorithmic Graph Theory*; Cambridge University Press: Cambridge, 1985.

(84) Gilson, M. K.; Honig, B. H. The dielectric constant of a folded protein. *Biopolymers* **1986**, *25*, 2097–2119.

(85) Caillet, J.; Claverie, P. Theoretical evaluations of the intermolecular interaction energy of a crystal: application to the analysis of crystal geometry. *Acta Crystallographica Section A* **1975**, *31*, 448–461.

Graphical TOC Entry