Classification of Protein Domain Movements using Dynamic Contact Graphs

Daniel Taylor

School of Computer Sciences
University of East Anglia

A thesis submitted in partial fulfilment for the degree of Doctor of Philosophy

November 2014

Acknowledgements

I would like to acknowledge my primary supervisor Dr Steven Hayward who has gone well beyond the call as a supervisor with his help and guidance of which I will be eternally grateful for. My secondary supervisor Gavin Cawley who came late to my supervision has helped me enormously grapple with some extremely complex topics throughout my study. I also enjoyed our casual chats over coffee in The Sainsbury Centre when time allowed. I owe my entire postgraduate academic success to you both.

I would like to highlight the care, help, support and friendship I received from Daniel Fuller and Oscar Pinnington over the last decade and more. I would also like to single out Robert William Turner whom I met at school 15 years ago and has been a truly “multibono” friend throughout those years who even now continues to cheer me up and make me laugh whenever it is required.

I would also like to acknowledge my very beautiful girlfriend, Miss Victoria Cartwright who has had an enormously powerful, positive and wonderful transformation to my life in recent years, whom I love and think the world of. Recognition needs to go to my fellow PhD comrade Alex Utev who I would often chat and have coffee with, who worked tirelessly in the field of protein inverse kinematics where I hope his work will be realised and published one day.

Finally and most importantly I would like to thank my entire family Barbara, John and Michael Taylor. I also include my grandmother Lottie Brown for the love, support, kindness and help they have given me over my entire life I would never have been able to get this far without you.

Abstract

Protein domain movements are of critical importance for understanding macromolecular function, but little is understood about how they are controlled, their energetics, and how to characterize them into meaningful descriptions for the purpose of understanding their relation to function. Here we have developed new methods for this purpose based on changes in residue contacts between domains. The main tool used is the “Dynamic Contact Graph” which in one static graph depicts changes in contacts between residues from the domains. The power of this method is twofold: first the graphs allow one to use the algorithms of graph theory in the analysis of domain movements, and second they provide a visual metaphor for the movements they depict. Using this method it was possible to classify 1822 domain movements from the “Non-Redundant Database of Protein Domain Movements” into sixteen different classes by decomposing the graphs for each individual protein into four elemental graphs which represent the four types of elemental contact change. For each individual domain movement the output of this process provides the numbers of occurrences of each type of elemental contact change. These were used as input for logistic regression to create a predictor of hinge and shear using assignments for these two mechanisms at the "Database of Macromolecular Movements". This predictor was applied to the 1822 domain movements to give a tenfold increase in the number of examples classified as hinge and shear. Using this dataset it was shown that contrary to common interpretation there is no difference between hinge and shear domain movements. The new data is presented online with new websites which give visual depictions of the protein domain movements.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Proteins are intricate molecules which play a central role in all biological processes. They operate in cells and are required for the formation of structure, for function, and for regulation of the body's tissues and organs. They can be defined according to their range of functions; examples are antibodies such as Immunoglobulin G (IgG), which bind to particular foreign agents such as virus and bacterial proteins, in order to protect the larger organism. Enzymes catalyse the numerous chemical reactions that take place with the cell. They help create new molecules by processing genetic information held within DNA. Messenger proteins, such as hormones, communicate signals to synchronise biological function between other cells, tissues, and organs. Transport/storage proteins such as ferritin bind and transport atoms and small molecules within cells and throughout the organism and structural proteins such as actin help to assemble the cell and maintain its integrity.

Many proteins can be subdivided into domains. A protein domain is the basic building block of protein structure, although many proteins only contain a single domain. Domains often have a distinct role, having a particular function or interaction. An example is the "Rossmann-fold domain" which has the role of binding coenzymes such as NAD. Although domains are often characterised by their function they may exist in a variety of biological contexts, occurring in proteins with different functions. A domain may be characterized as

being a spatially separated unit of protein structure. It may have sequence and/or structural resemblance to a domain in a different protein with which it may have related function. Domains in multidomain proteins often have the potential to move relative to one another and this movement is often intimately involved in function. These protein domain movements are the subject of this study. Structures of proteins are determined by X-ray crystallography which offer "snapshots" of proteins in conformations that often relate to distinct functional states. Thus the study of conformational changes in proteins can provide insight into mechanism.

Protein domains move as quasi-rigid bodies and various methods based on rigid-body kinematics have been developed for their analysis of these movements. These methods require at least two structures for input. They identify domains and describe their relative movement by way of hinge axes about which the domains rotate relative to each other. In this work the DynDom program has been used [49]. From the basic input of two structures DynDom's output are the domains, as characterised by the amino acid residues that they comprise, the hinge axes, and hinge bending residues. DynDom output from the analysis of a large number of proteins forms the basis of this study. The aim of the study is to characterise domain movements putting them into different classes based on contacts made and broken during the movement.

We have developed a new method for analysing and visualising protein domain movements based on domains identified by DynDom and the changes in contacts between residues from different domains. A new directed graph called the "Dynamic Contact Graph" (DCG) is proposed that has a number of strong features. First, in one graph it encapsulates both sets of contacts, that is, contacts between residues in both structures. This has the advantage of allowing one to see in one static depiction changes in contacts due to the movement. Second, they provide a visual metaphor for the movement they represent. Third, they allow the powerful methods developed for the analysis and display of graphs to be utilised. The DCG represents the main breakthrough in this work. From there everything else follows.

Using DCGs a large number of domain movements have been classified into 16 types based on decomposing their DCGs to determine the number of instances in each of one of the four different types of basic contact changes that exist. This represents a new and unique way of classifying protein domain movements.

The initial motivation for this work came from the attempt to classify domain movements according to Gerstein and co-workers at Yale University. They saw two main types of mechanisms at work, shear and hinge, and they classified domain movements into the two mains classes: *predominantly* shear and predominantly hinge. Domain movements were assigned to these classes largely using subjective methods. Our motivation was to develop an objective and automatic method for the assignment of domain movements into these two broad classes. Although the DCG provided us with an alternative classification method, we extended our DCG work to classify domain movements into the predominantly shear and predominantly hinge categories. This was achieved using a machine learning approach which allowed us to classify, in an objective way, a tenfold larger set of domain movements than previously, into predominantly shear and predominantly hinge classes.

This thesis starts with a background literature review, setting the scene for the introduction of the concept of the DCG. The background section begins with an explanation into what proteins are and how their structures are determined experimentally. . The background section also describes methods used to characterise them based on their structure and introduces the various methods used to determine domains. It also describes the current methods used for domain movement analysis. The methodology section introduces all the methods needed to understand the techniques and calculations used in our research and includes basic set theory, basic graph theory and regression techniques. The results section is split into three sections following a chronological narrative allowing the reader to follow the path taken. The first part of the results section covers the residue-based contact method; a method that was found to be inadequate for the description of certain types of contact changes. This work motivated

us to create the DCG concept. The second part presents the DCG work itself and the third part presents results of applying the results from the DCG analysis to the prediction of hinge and shear mechanism using logistic regression. The thesis concludes with the discussion which looks at the potential future use of the DCG method.

Chapter 2

Background

2.1 Proteins:

Proteins are large dynamic macromolecules, vital for biological function. They are involved in important and various tasks within the cell. The way these macromolecules are formed and function is of huge importance and interest. Once fully folded, the internal movements they make to perform their functional purpose are enormously relevant. It is also their function which classifies them by name. The numerous roles proteins have include providing structural strength, such as in hair and fingernails, controlling the immune system and metabolism, transporting other molecules around the body and processing various organic agents from one molecule into another as a product. Therefore a protein's function and structure can be seen as interdependent as they are reliant upon each other. A protein is a composite macromolecule produced from a string of amino acids linked by peptide bonds, known as the polypeptide chain, which when folded forms a three-dimensional structure determined by its polypeptide amino acid sequence. There is a hierarchy which describes the protein structure: primary, secondary, tertiary and quaternary [12].

2.1.1 Protein Primary Structure:

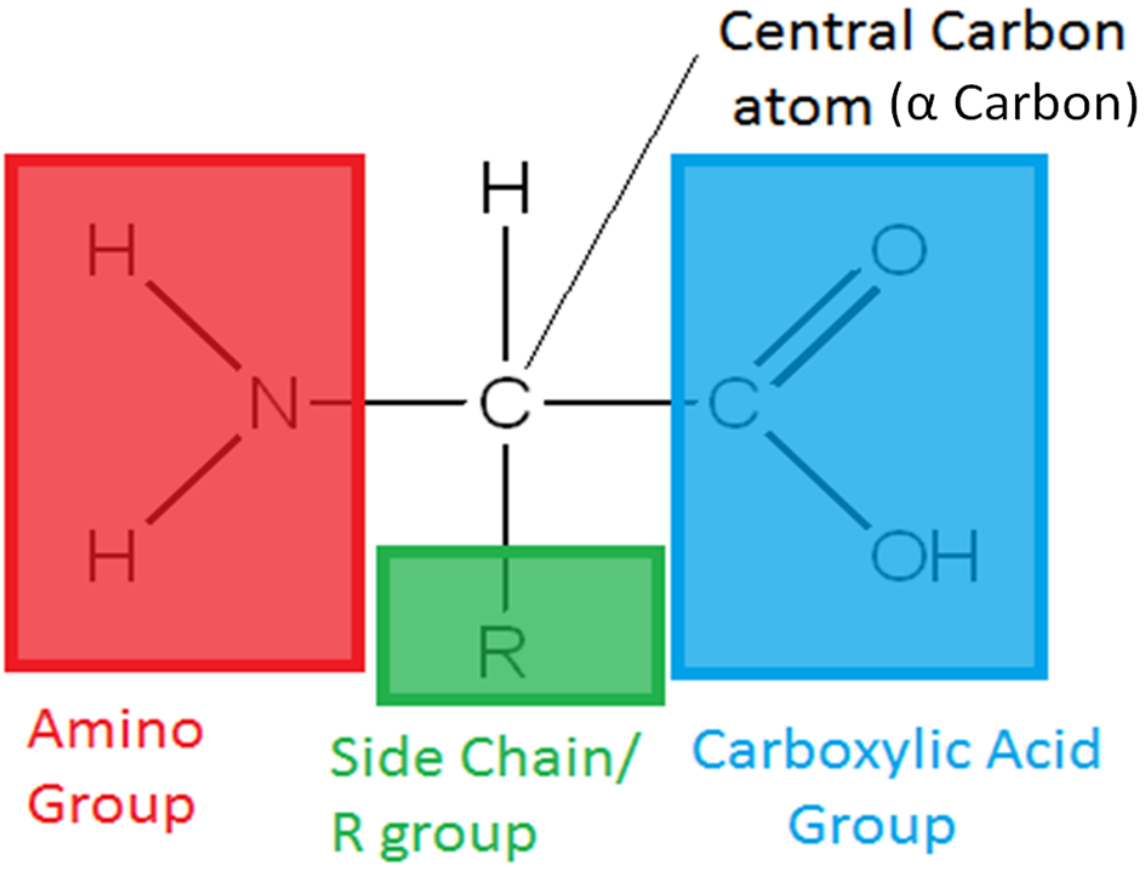

Fig. 2.1 Basic structure of an Amino Acid

The primary structure is the amino acid sequence. This sequence is encoded by a gene, a portion of DNA within the cell nucleus. There are 20 naturally occurring amino acids, all of which have a central carbon atom (Cα); this joins together a carboxyl group (COOH) and an amino group (NH2). Attached to the Cα atom is a hydrogen atom and a side chain, often referred to as an R group, which confers identity to the amino acid (Figure 2.1). The peptide bond is formed between the amino and carboxyl group (Figure 2.2) during protein synthesis.

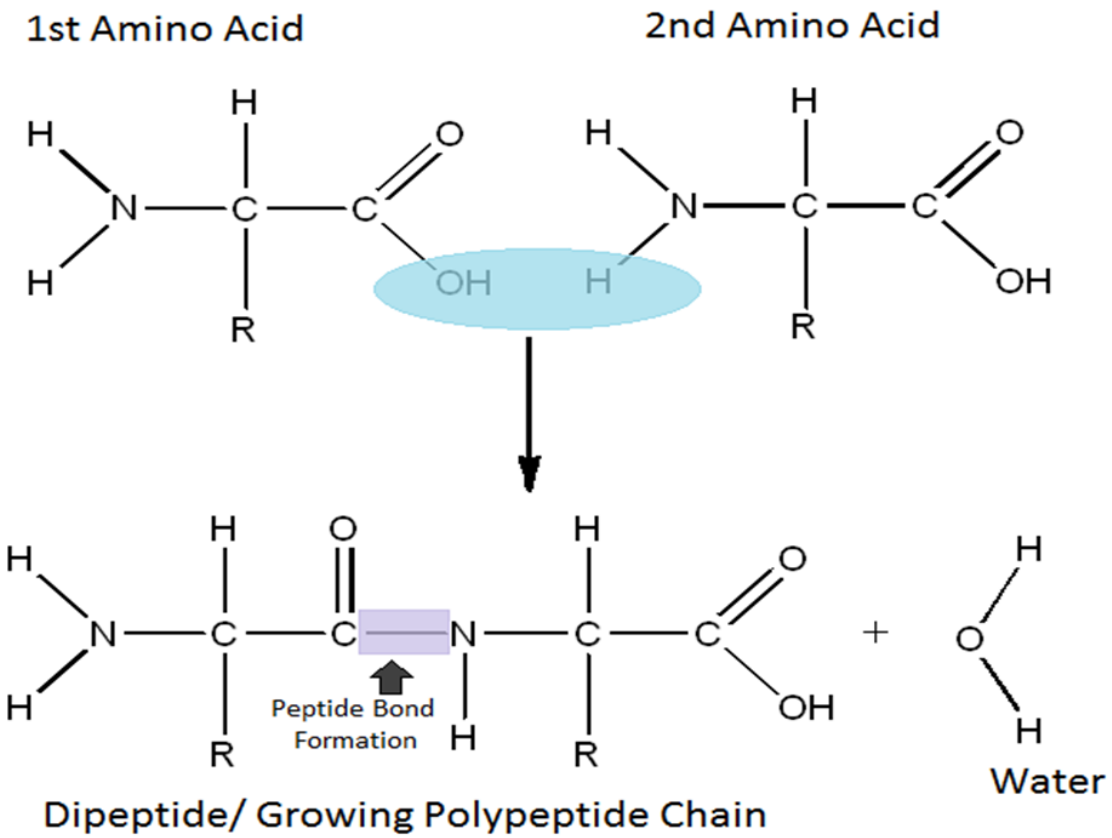

Fig. 2.2 Peptide Bond Formation

The amino acids can be characterized by name; because of their differing chemical structure, each amino acid can be grouped with others according to whether it is basic or acidic, hydrophilic or hydrophobic, polar or aromatic. Their full names are often abbreviated by a single letter or a three letter code (Figure 2.3).

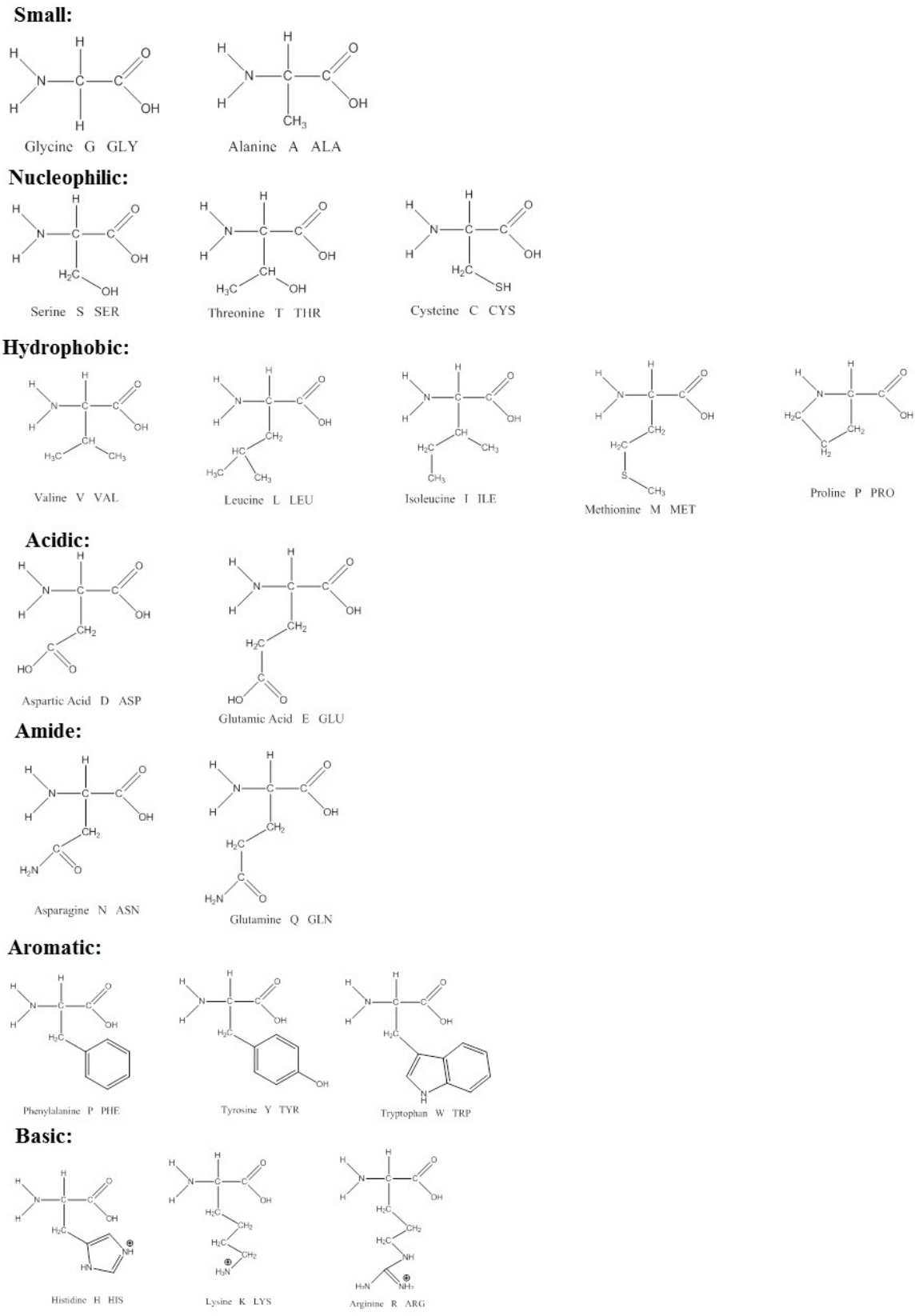

Fig. 2.3 Twenty natrually occuring Amino Acids (Acidic and Basic both relate to charge) (Alanine is also considered Hydrophobic) (Nucleophilic, Amide and Aromatic (except PHE) relate to polar) Glycine (G) has a single hydrogen atom as a side chain so can be classified as either hydrophobic or separate from the other amino acids

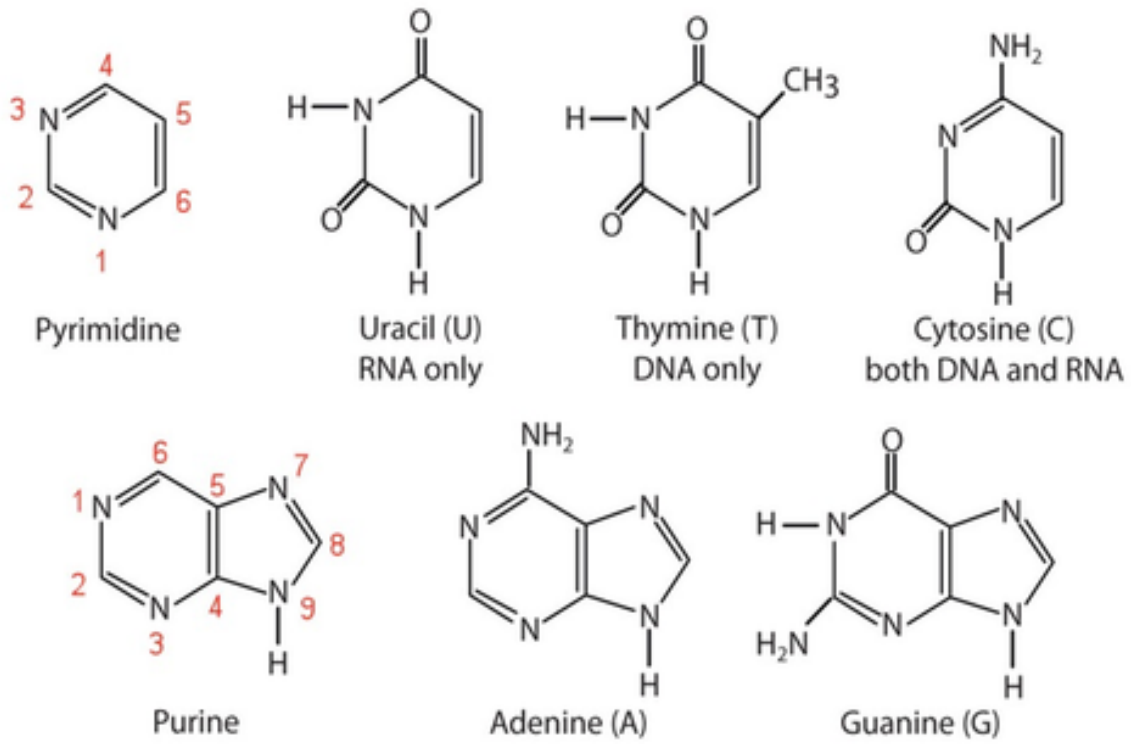

Fig. 2.4 DNA & RNA base codons (please note Thymine (T) only exists in DNA and Uracil (U) in RNA but are interchangeable with one another)

The blueprint necessary to construct proteins is found in DNA, coded by a sequence of 4 bases; these can be further subdivided into 2 categories, Pyrimidine (a single 6 membered heterocyclic organic compound) and Purine (6 membered joined to a 5 membered heterocyclic organic compound). These bases form triplets (codons) in the genetic code (Figure 2.4). Once the gene encoding the sequence of the polypeptide is “transcribed” to mRNA (messenger RNA) it has to be then "translated" into the "language" of the protein. The sequence of bases in the mRNA molecule forms a code, just as in the DNA molecule, for the amino acid building blocks of the protein. The information in the DNA is thus translated from the four character alphabet of bases to the twenty-character alphabet of amino acids. The production of proteins takes place on ribosomes, which are attached to the endoplasmic reticulum (which controls the transportation system of the cell). The ribosome reads the sequence of codons on the mRNA molecule and matches each codon to its anticodon on a tRNA molecule, which

brings with it the corresponding amino acid (Figure 2.5).

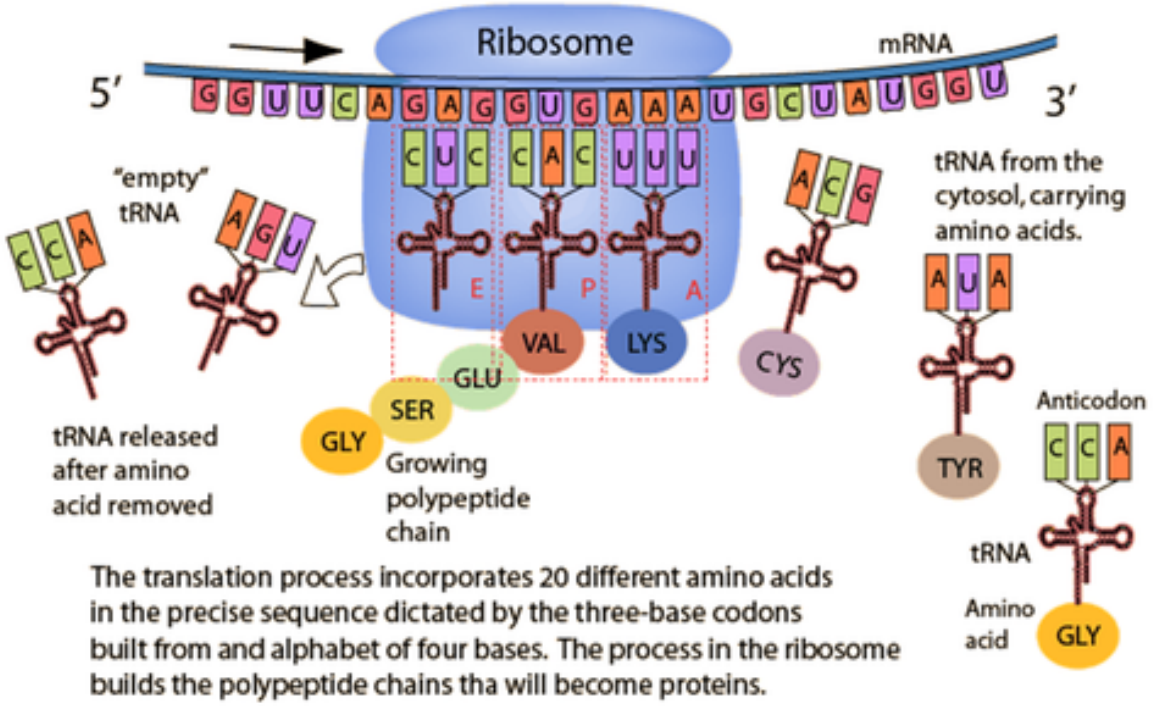

Fig. 2.5 Translation Process in Polypeptide (Protein) construction [85]

The translation process builds a polypeptide with the specific sequence of amino acids specified by the mRNA molecule. The method of translation can be separated into the stages of initiation, elongation, and termination. Initiation incorporates at least three other proteins called initiation factors to aid mRNA binding to the lesser subunit of the two-unit ribosome. It is bound to the correct location using the initiation codon AUG (the start codon) on the mRNA. The next phase is elongation, with the addition of other amino acids to the elongating polypeptide chain and finally termination, when translation comes to an end and the ribosome dissociates.

2.1.2 Polypeptide Geometry:

Amino acids assemble by the formation of peptide bonds where they are joined end to end. The amino group of the first amino acid and the α-carboxyl group of the last amino acid remain unchanged. The polypeptide chain lengthens from its amino terminal (N-terminal) to its carboxyl terminal (C- terminal) in a repeating pattern (excluding the side chains); this gives the protein's "backbone" or "main chain" ($NH-C\alpha H-C'=O$) which is integral to the protein's flexibility and internal motions. The backbone gives rise to three torsion rotations about three repeating bonds ($N-C\alpha, C\alpha-C', C'-N$) which together play a critical role in the determining a protein's folding and its native three-dimensional structure (Figure 2.6). These rotations about each bond are quantified by torsion angles defined by the four backbone atoms or dihedral angles defined as the angle between two planes formed by two overlapping triplets of backbone atoms. The rotation around the $N-C\alpha$ bond is defined by the torsion angle Phi (ϕ) $C'-N-C\alpha-C'$.

The rotation around the bond $C\alpha-C'$ is defined by the torsion angle Psi (ψ) $N-C\alpha-C'-N$. The path of the polypeptide backbone is largely determined by these two angles [93]. The rotational angle of the peptide bond, defined by $C\alpha-C'-N-C\alpha$ is called omega (ω). These four atoms are inclined to be planar because of the delocalized conjugated pi-system from the $C'=O$ bond over the peptide group. The inhibition of rotation around the $C'-N$ bond, gives rise to the rigid peptide plane from Cα to the next Cα along the main chain backbone. Conversely the single bonded $N-C\alpha$ and $C\alpha-C'$ which connect the peptide units can freely rotate as long as there is no steric limitation from side chains.

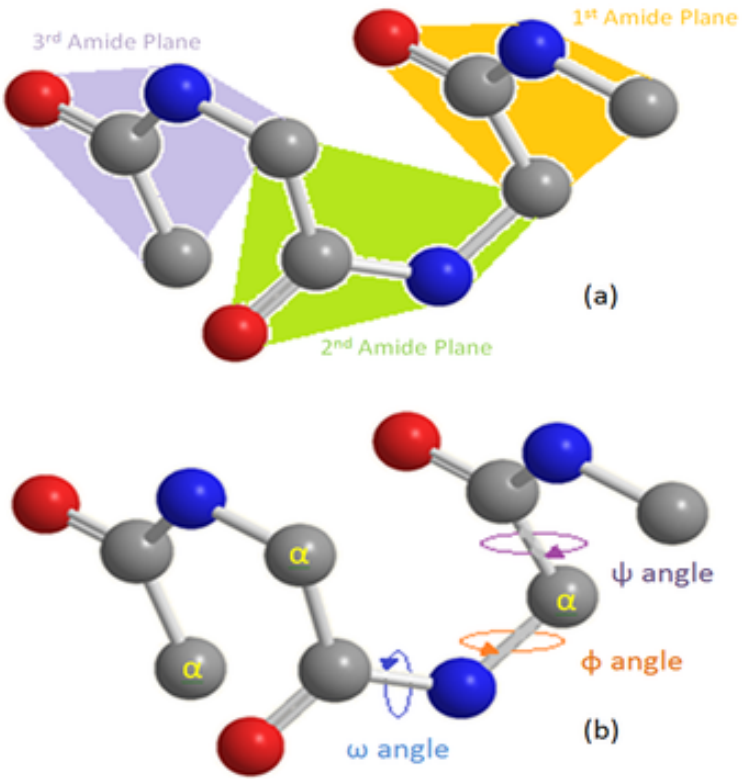

Fig. 2.6 Protein backbone diagram with R group side chains and hydrogen's omitted (a) Amide Planes in Polypeptide (b) Torsion angles in the Polypeptide Backbone Phi (ϕ) Psi (ψ) and Omega (ω). Omega is not free to rotate (nearly always 180 degrees)

2.1.3 Protein Secondary Structure:

α-Helix:

The next level in the hierarchy of protein structure is defined by the hydrogen bonding of the backbone with itself to form "secondary structures". The two most common secondary structures are the α-helix and the β-sheet; these differ from one another due to their hydrogen bonding patterns. The α-helix is a tube like right-handed helix, with the side chains extending out from the helical axis (Figure 2.7). There are 3.6 amino acids per turn, with one turn rising 5.4Å along the helical axis. Hydrogen bonds are formed from the CO of one residue (i) to the NH group four amino acids away (i+4) giving this secondary structure inherent local

strength.

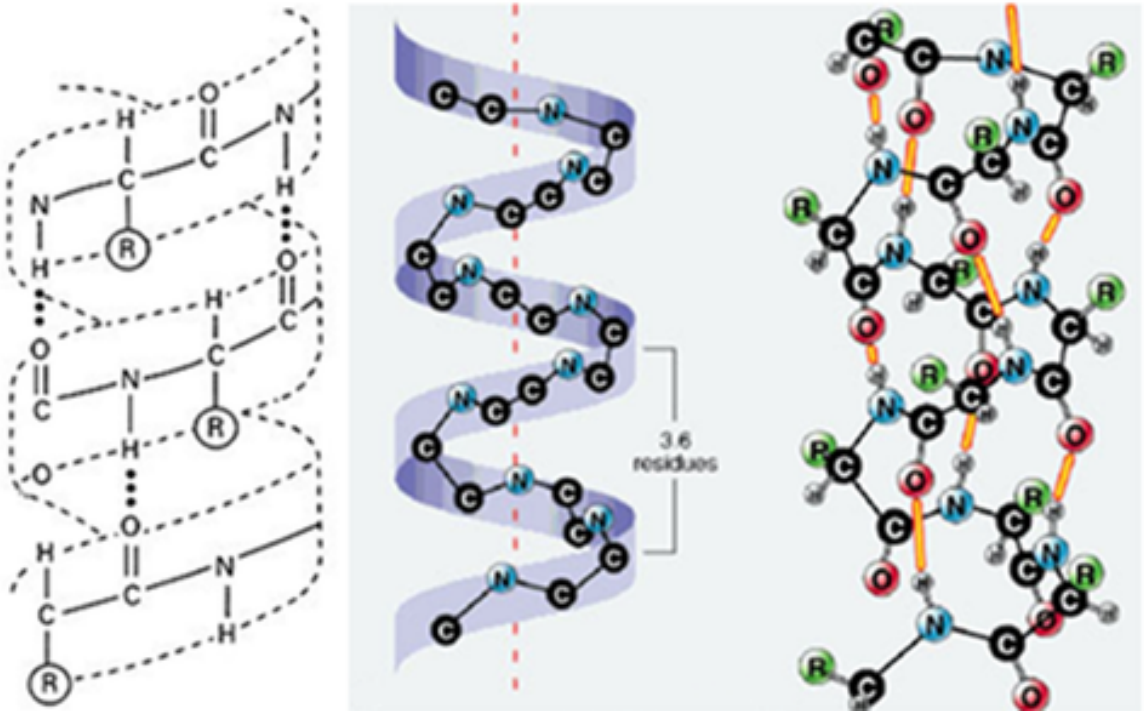

Fig. 2.7 Hydrogen Bonding of The Alpha Helix [116, 133]

β-Sheet:

The β-sheet secondary structure is a collection of strands of backbone, commonly 5 to 10 amino acids in length, aligned side-by-side. The hydrogen bonding is between the CO on one β-strand to the NH group on the neighboring β-strand and vice versa. The side chains R groups point alternatingly above and below the plane of the β-sheet. The Cα atom sits alternatively slightly higher and lower in the plane of the sheet due to the restrictions of the hydrogen bonding giving rise to a pleated structure. There are two varieties of β-sheet: the parallel or antiparallel β-sheet. If neighboring strands point in the same direction, it is parallel β-sheet; if neighboring β-strands point in opposite directions it is an antiparallel β-sheet (Figure 2.8).

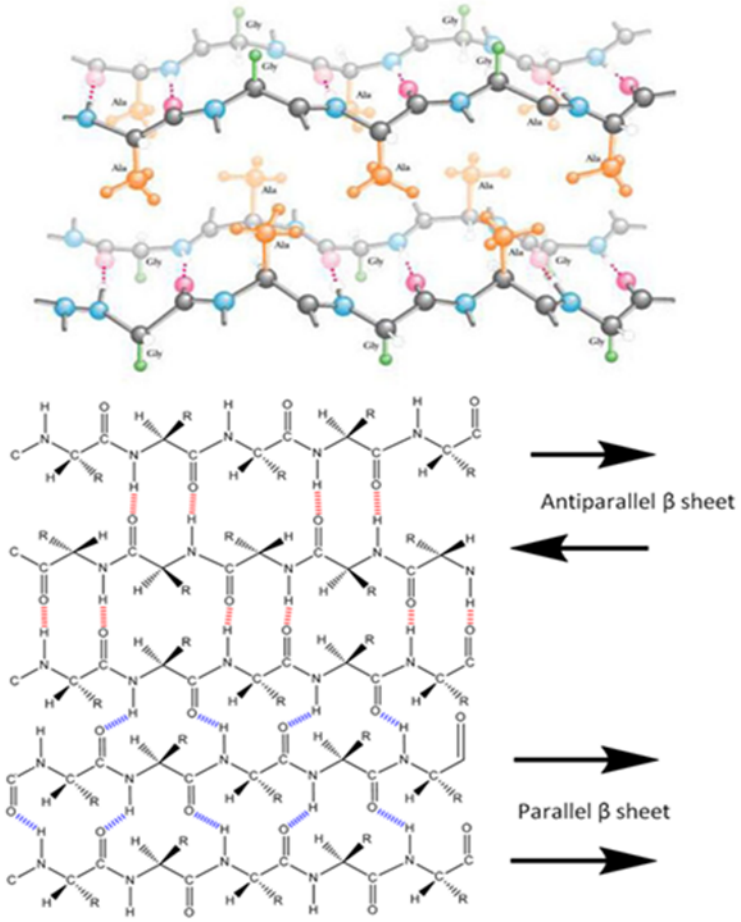

Fig. 2.8 Hydrogen Bonding of β-Sheet Secondary Structure [20, 104]

2.1.4 Protein Tertiary Structure:

Tertiary structure refers to the overall three-dimensional arrangement of atoms in a single folded polypeptide chain. This arrangement is stabilized by the packing of side chains within α and β secondary structures. The arrangement of secondary structures within a protein is often referred to as a “fold” (Figure 2.9).

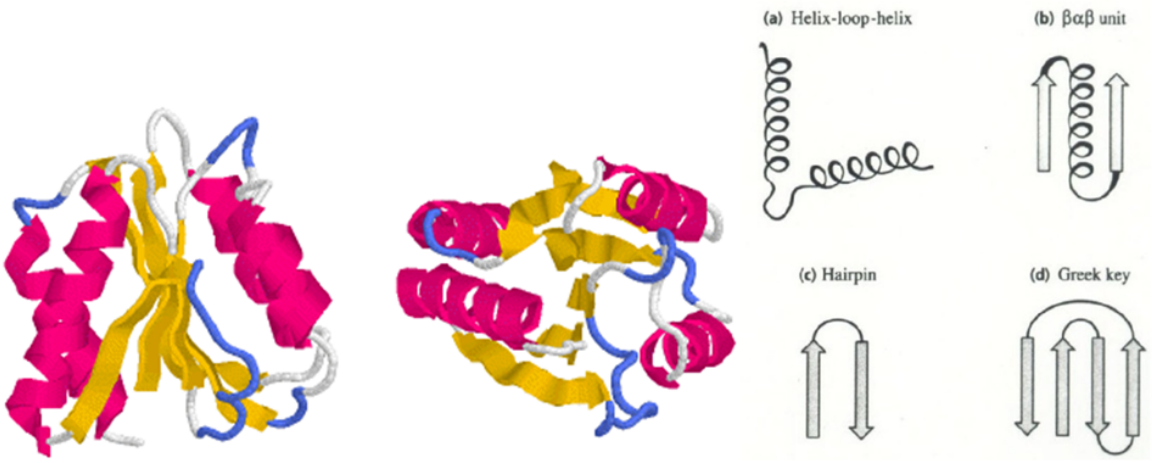

Fig. 2.9 Tertiary/Super Secondary Structure of Protein Folds [6, 86]

Folds are defined at the level of domain, compact globular features that are thought to be able to fold independently of the rest of the protein. Folds are further classified into three key structural classes; α-structures (made entirely from α secondary structure), β-structures (made entirely from β secondary structure) and α-β structures which can be further subdivided into α/β (with a discontinuous α and β organization, mostly with parallel β strands) and α+β (with a more clear-cut division between the α and β secondary structures, with the β strands being largely antiparallel). These produce essential structural and functional components of the tertiary structure.

2.1.5 Protein Quaternary Structure:

A protein often comprises more than one polypeptide chain. These individual folded chains are termed “subunits”. Quaternary structure refers to the arrangement of subunits (if it has two subunits, it is known as a dimer, three is a trimer, and so on). Usually these subunits interact with one another via non-covalent bonding (Figure 2.10).

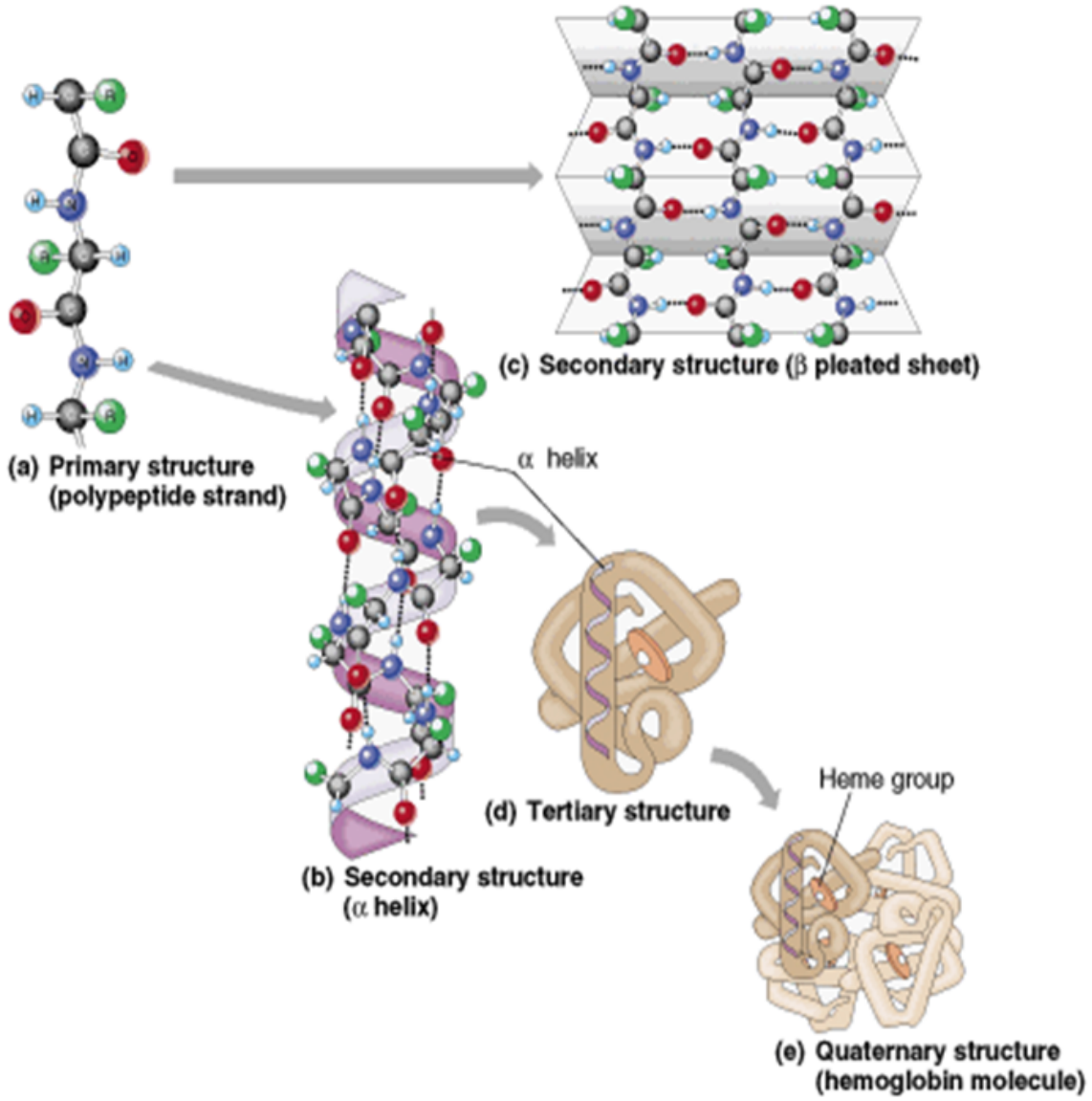

Fig. 2.10 Hierarchy of Protein Structure, from Primary up to Quaternary Structure [6, 86]

2.2 Determination of Structures:

Our knowledge of protein structures comes primarily from X-ray crystallography [13] and Nuclear Magnetic Resonance (NMR) experiments [88]. These give atomic level information on protein structure as well as potential internal movements. This structural data is stored at the Protein Data Bank [10].

2.2.1 X-ray Crystallography:

X-ray crystallography takes a sample of a specific protein and crystallizes it. A crystal contains large quantities of identical unit cells, packed alongside one another in a three-dimensional array. Each unit cell contains at least one molecule but often more than one, in which case they can be related to one another by symmetry. The "asymmetric unit" is the smallest portion of the crystal from which it can be created by symmetry operations. An asymmetric unit can contain one, a portion of, or multiple molecules (Figure 2.11) and does not necessarily contain the biologically active molecule (known as a biological unit). The symmetry operations required for generation of the whole unit cell involve translations, rotations and screw movements [102].

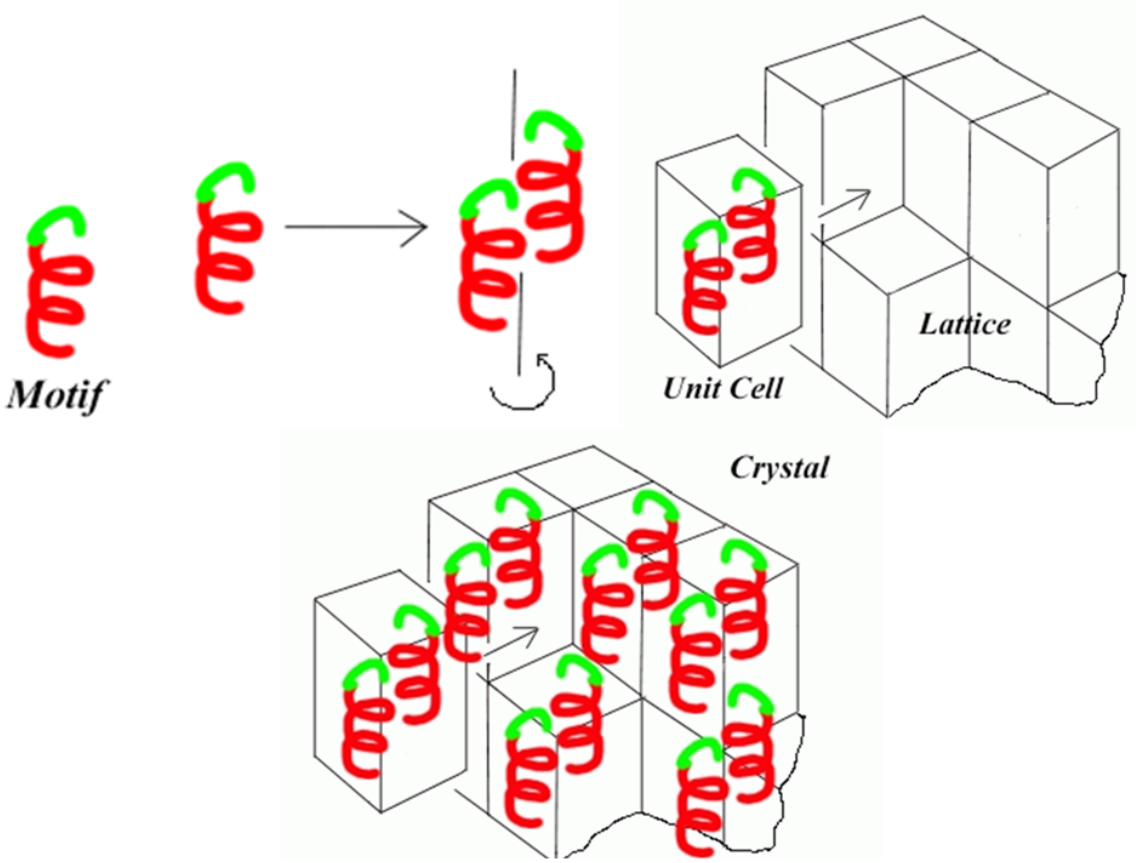

Fig. 2.11 Symmetric/Asymmetric units producing a Unit Cell, packing into a 3D Lattice Array [103]

The crystal will diffract an X-ray beam due to the electrons in the atoms to produce a diffraction pattern. The diffraction pattern comprises a series of spots corresponding to regions of high X-ray intensity. Through the use of Fourier transforms, the diffraction pattern can be converted into a three-dimensional electron density map (Figure 2.12). The polypeptide chain is fitted into this electron density map, in a process called refinement to eventually give the atomic coordinates of atoms within the molecule [102]. This coordinate data is stored in a PDB (Protein Data Bank) file.

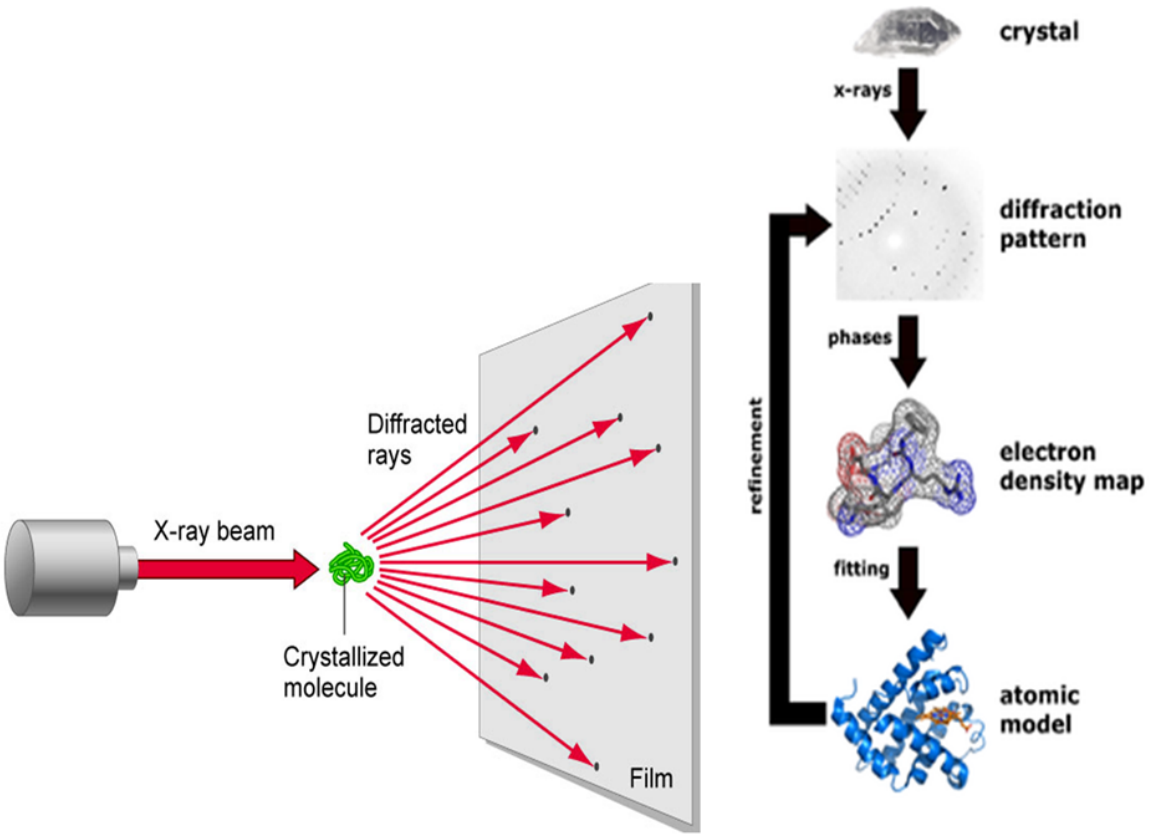

Fig. 2.12 X-ray Crystallography diffraction and atomic model generation through electron density mapping [73]

The fine detail and quality of the final atomic structure is dependent upon the resolution of the electron density map, which in turn is reliant on the resolution of the diffraction pattern which is amongst others contingent on the quality of the crystal. The higher the resolution, the more superior the final atomic structure will be and the less prone to error; for example a resolution of 5Å produces only a general envelope of the protein whereas a resolution of 1.5Å allows individual atoms to be seen very clearly (Figure 2.13). [12, 32, 98].

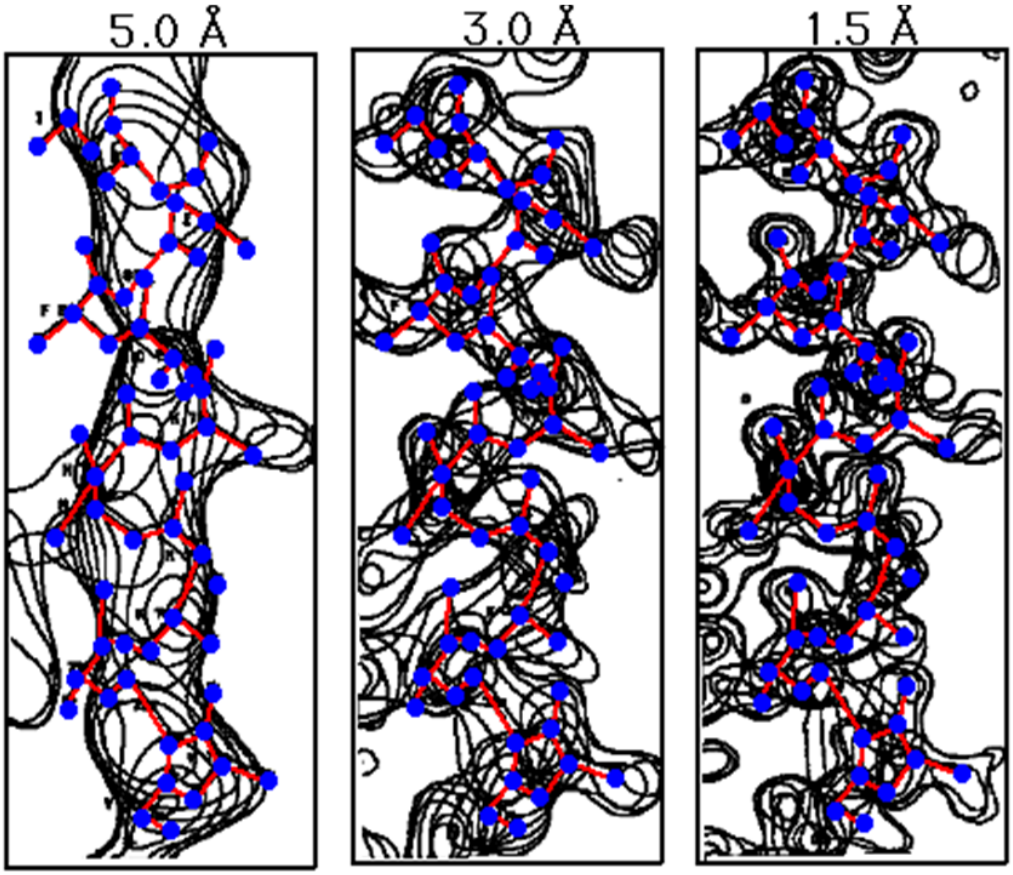

Fig. 2.13 Electron Density Maps at 5.0, 3.0 and 1.5Å [52]

2.2.2 Protein NMR:

Spectroscopy has provided a valuable insight into investigating protein structure. Nuclear Magnetic Resonance (NMR) spectroscopy uses the electromagnetic spectrum with a sample of the protein in a magnetic field to manipulate hydrogen H^1 (which is most abundant), carbon C^{13} and nitrogen N^{15}, the latter two of which do not occur naturally, to control their nuclei spin states. The chemical environment can be further explored and the distances between atoms measured. A three-dimensional model can be created from these results [12]. Unlike X-ray crystallography, NMR has the advantage of not requiring a crystal; it can also determine the positions of individual hydrogen atoms in the structure. Protein NMR is, however, restricted to comparatively small molecules [32].

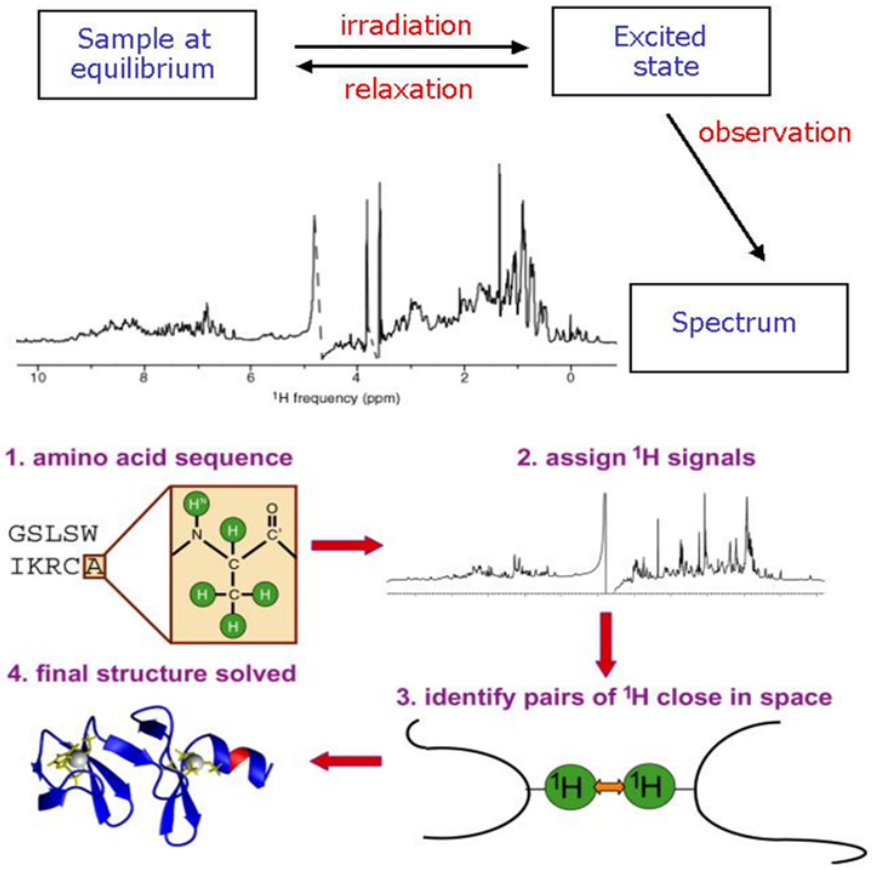

Fig. 2.14 Protein NMR [77]

When placed in a magnetic field the nuclear spin of the H^1, N^{15}, C^{13} atoms align to the field. The sample is then irradiated with radio waves (Figure 2.14). The subsequent relaxation produces a radio wave whose properties (frequency, intensity, relaxation time) give information on the local environment of the nuclei and their dynamics. Frequencies give information on chemical shifts which are directly related to the identity of the nuclei and their local magnetic field. In proteins the chemical shifts alone cannot discriminate particular atoms, as overlapping peaks can combine to give a single peak. Two-dimensional NMR experiments are used to counter this problem. In 2D NMR experiments two pulses of radio waves are applied separated by a relaxation period. At the end of the experiment the data can be plotted, defined by two frequency axes rather than one.

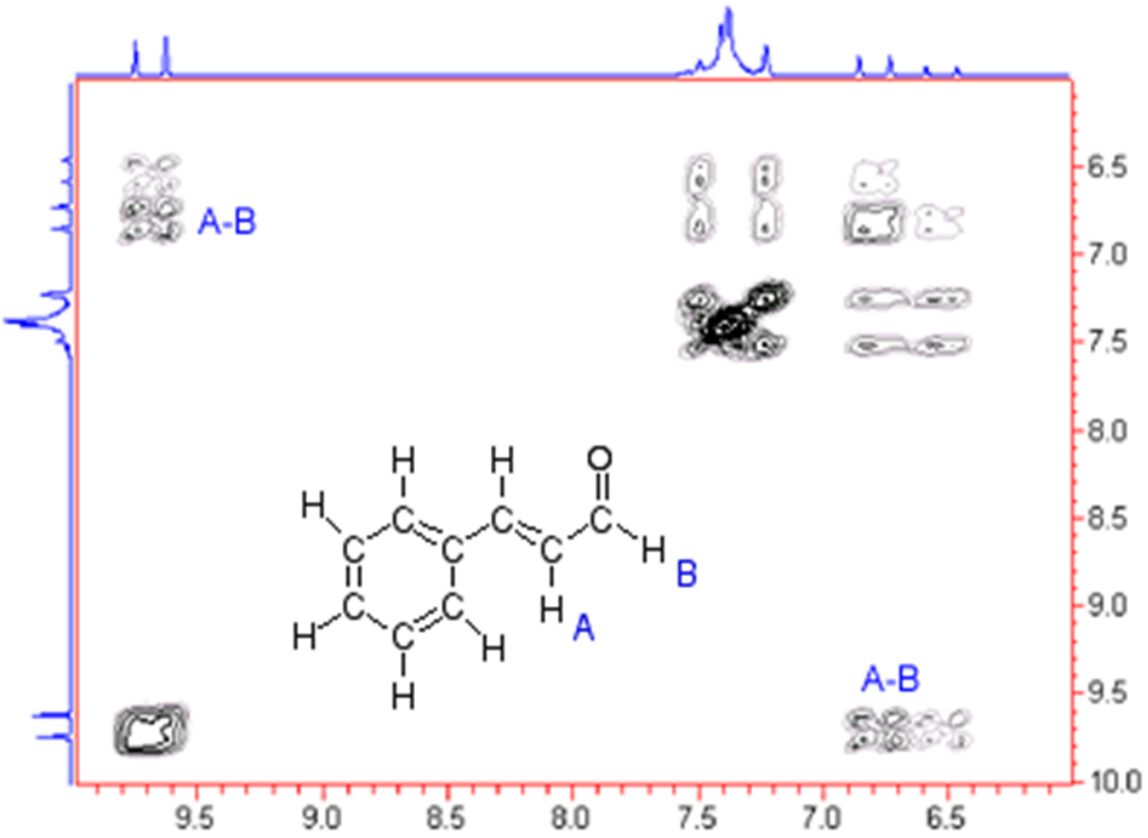

Fig. 2.15 COSY NMR Protein Spectroscopy Reading [67]

The two main types of two-dimensional NMR experimentation are COSY (Correlation Spectroscopy), which highlights the peaks between atoms connected by bonding (Figure 2.15), and NOESY (Nuclear Overhauser Effect Spectroscopy), which gives peaks between pairs of atoms close to one another in space (Figure 2.16).

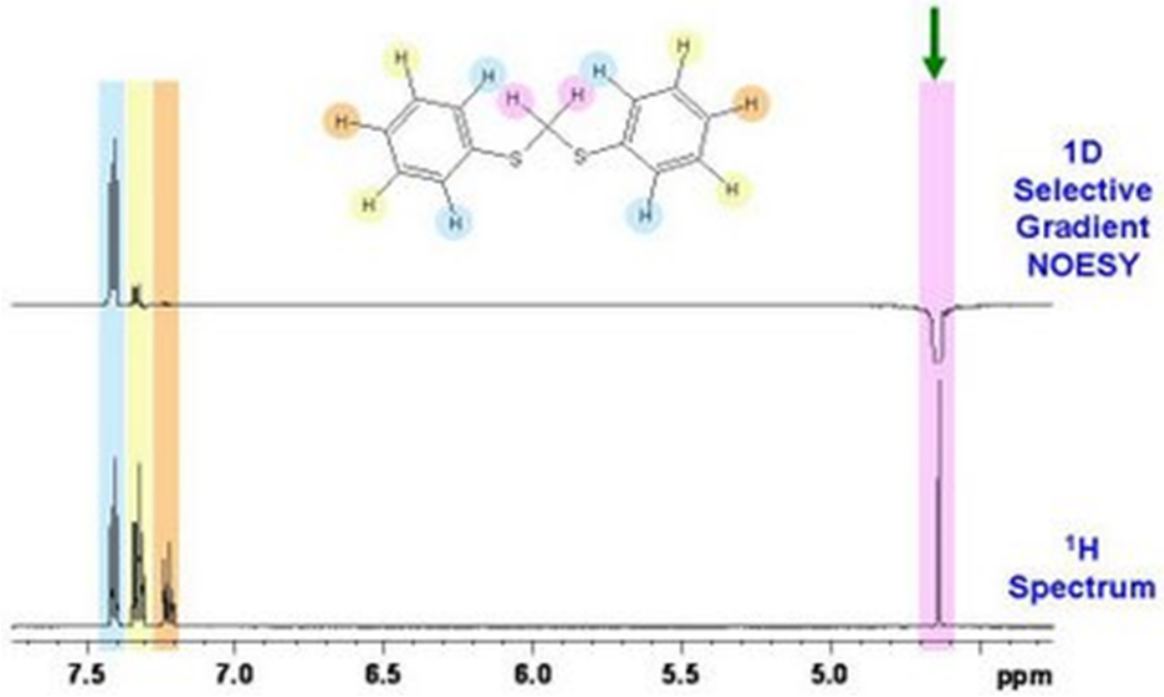

Fig. 2.16 NOSY NMR Protein Spectroscopy Reading [29]

The spectrum gives a set of approximate distances between pairs of hydrogen atoms. A three-dimensional model using the polypeptide sequence can be fitted to satisfy these distance constraints. NMR protein spectroscopy remains the method of choice for proteins not available for crystallization [102].

2.3 Protein domains:

Domains are the fundamental evolutionary unit of protein [94]. From a structural perspective domains can be defined as compact, quasi-globular regions. They may be able to fold independently (Figure 2.17). A domain is the smallest indivisible unit conferring function and can be regarded as transferable functional entities in evolution. Domains can appear as whole proteins or linked together; in the latter their relative movement can be important for function.

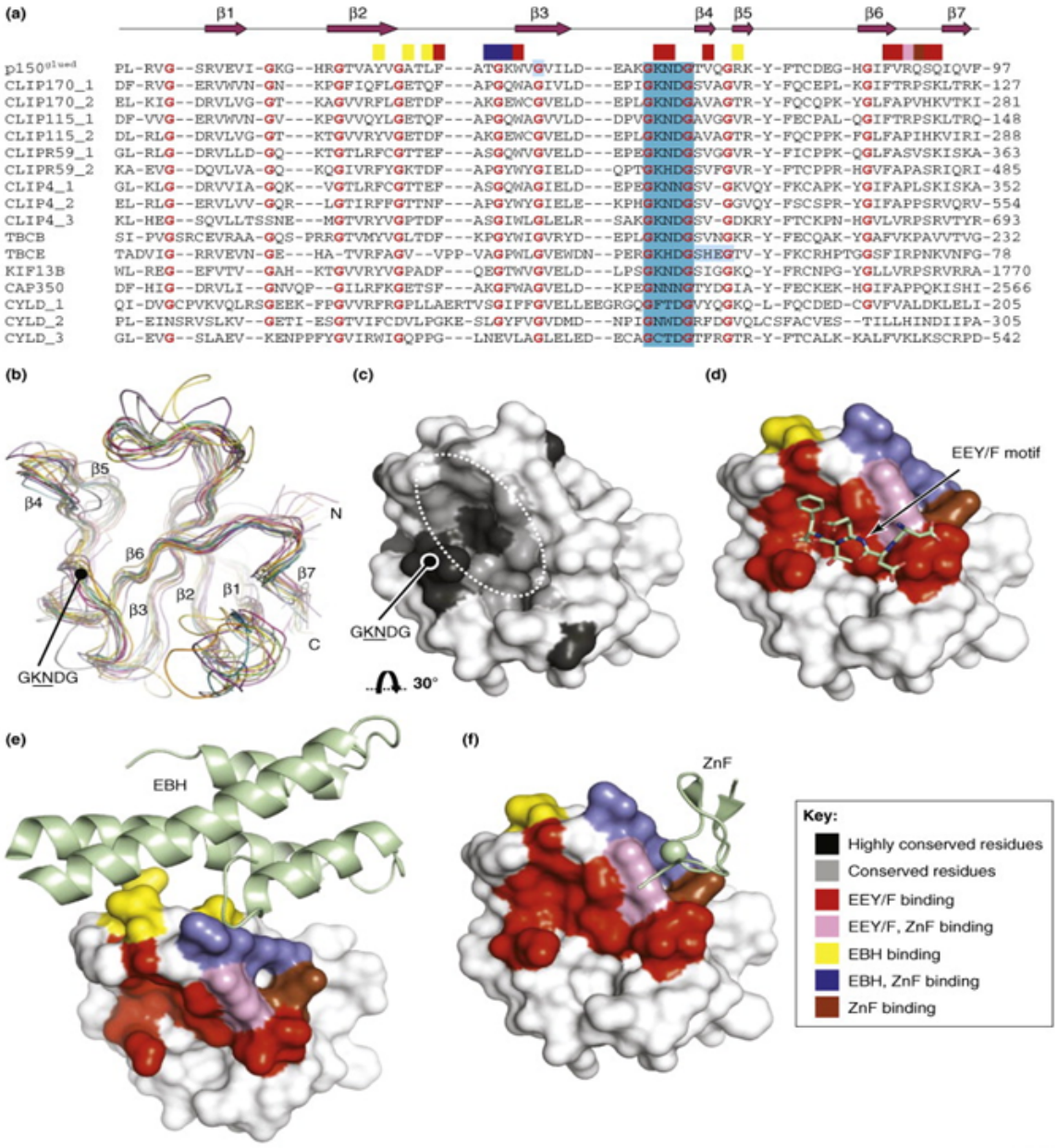

Fig. 2.17 Conservation of Sequence and Structure in Proteins highlighted by biological necessity [118]

2.3.1 Protein Domains Identified by Structure:

The exact definition of where one domain begins and another ends depends upon the method used (Table 2.1). Alcohol dehydrogenase (PDB code 18NK), for example, shows slightly different definitions for its two domains according to the two structure classification databases

SCOP [84] and CATH [89], discussed in (2.5).

	Domain A	Domain B
SCOP	1-163, 340-374	164-339
CATH	1-178, 318-374	179-317

Table 2.1 Comparison of Alcohol Dehydrogenase (18NK) between the two Domain Databases SCOP and CATH

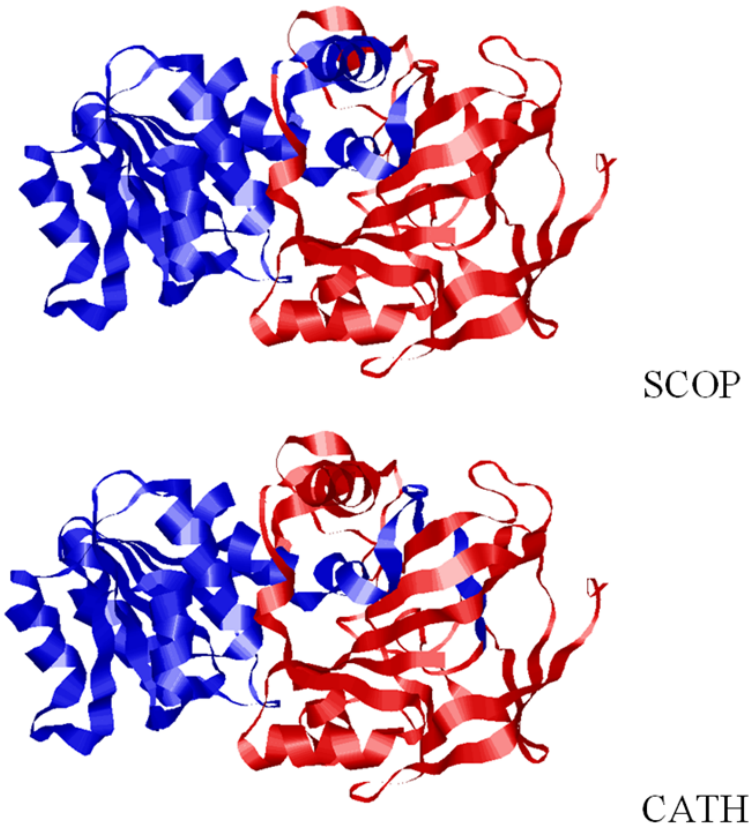

Fig. 2.18 Alcohol Dehydrogenase (1N8K) Structural Domains according to the SCOP and CATH databases

Domains can be classified according to structural similarity. Methods have therefore been developed to search for structural similarities, or to predict protein structure, at the domain level using fold recognition [24]. The detection of these domains in proteins of unknown structure can offer a valuable insight into the protein's function. If the biological and

functional character of a homologous domain is not known, the comparison of these regions could still uncover conserved residues as mutagenesis targets in wet lab experimentation.

2.4 Identification of Domains:

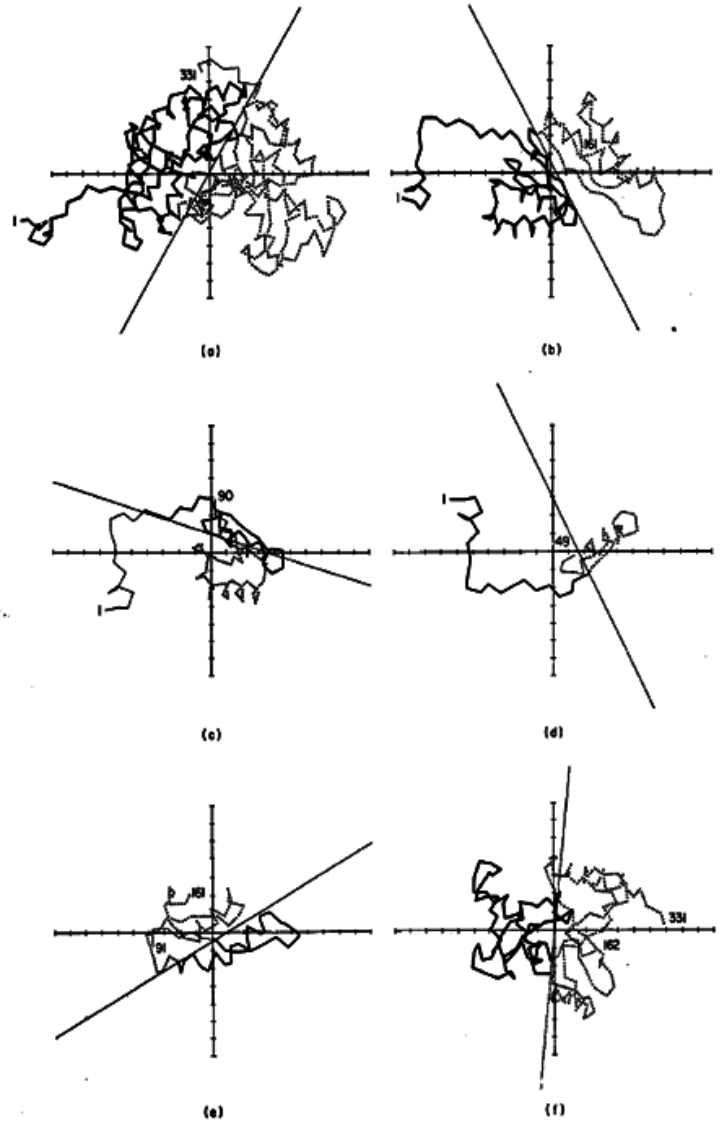

Fig. 2.19 3D Perpendicular Axis System for identifying domains, which travels along the polypeptide backbone analysing it according to its axis system [101]

There are a number of ways to characterize domains. Many large proteins can be regarded as a collection of domains. In such multi-domain proteins residue contacts will be more extensive within domains than between domains [61, 99]. Many methods that identify domains from structure use Cα to Cα distances [121, 135]. The method presented by Rose

[101] optimally divides the protein into segments hierarchically (Figure 2.19),(Figure 2.20). A quite different method was devised by Holm and Sander [56] by the use of contact matrices. This method is a sort of simplified Normal Mode Analysis identifying domains by determining residues that move concertedly.

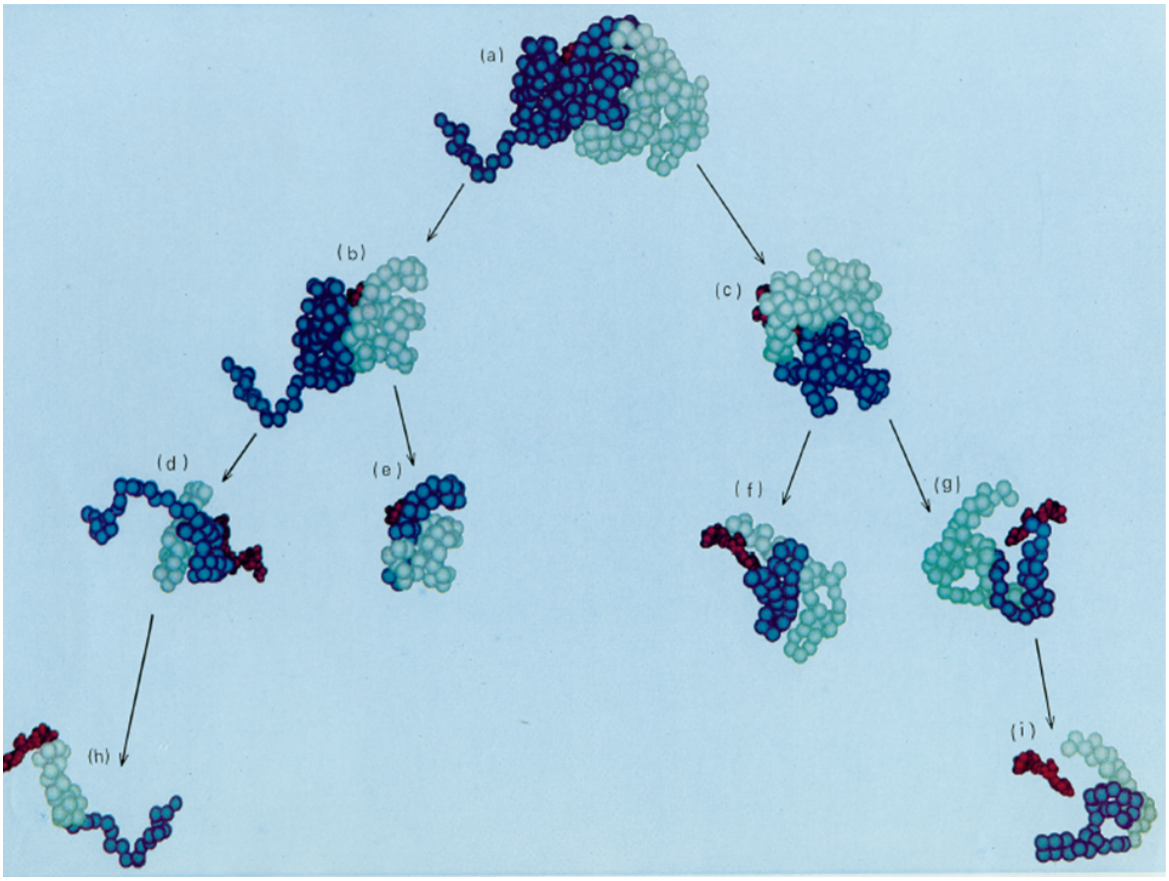

Fig. 2.20 Step-wise processing of the Domains starting with the complete protein and ultimately leading to separate domains, using computer generated space filling models [101]

Another method has been described by Siddiqui *et al.* [113]. The process involves partitioning the polypeptide backbone and evaluating the interaction energy between the divided units with the aim of keeping the number of partitions and the interaction energy as low as possible (Figure 2.21). The solvent accessible area is a main contributor to the interaction energy. If a split of the polypeptide backbone does not affect the solvent accessible area, then it would label a domain division at this point [131].

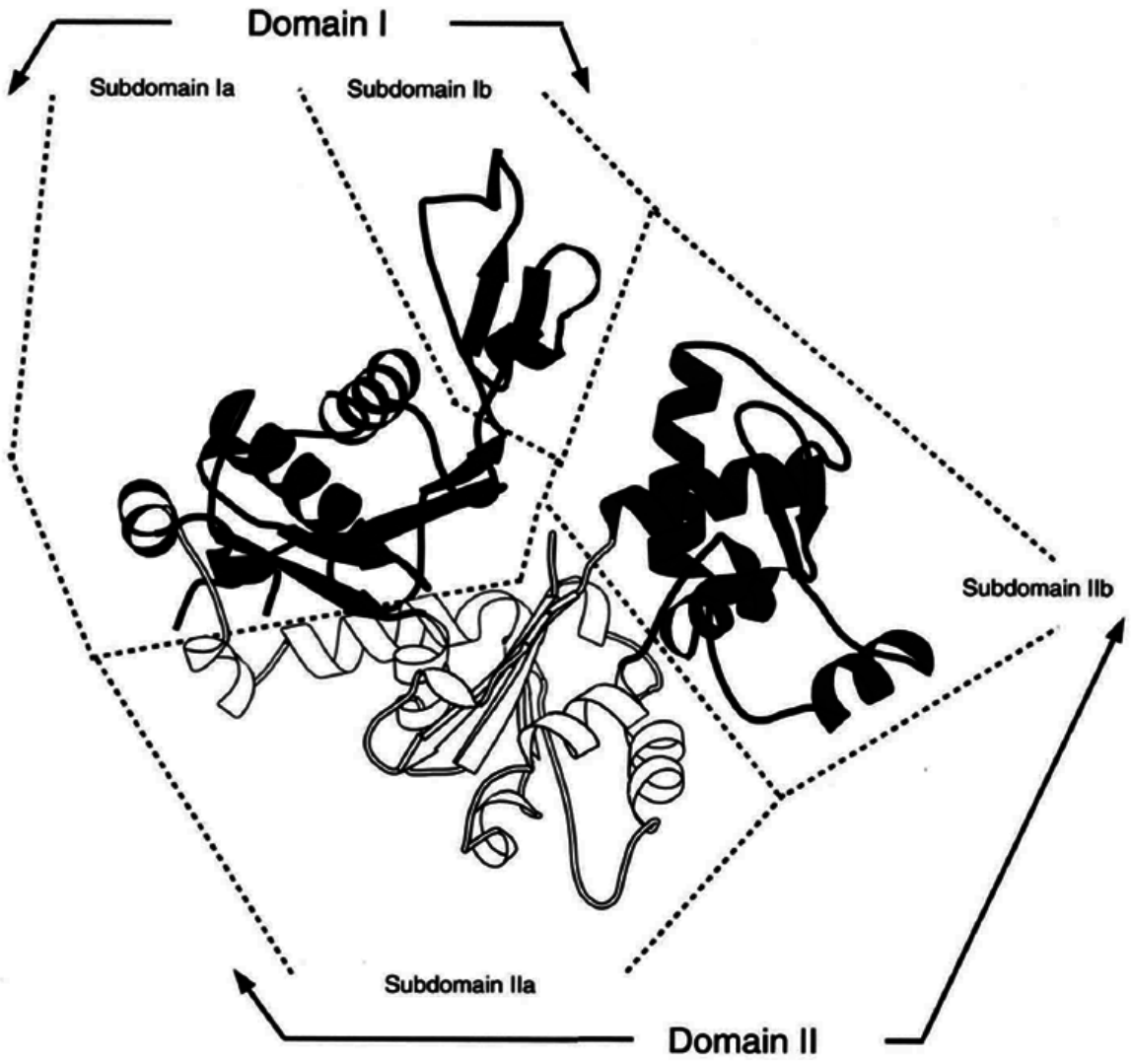

Fig. 2.21 Example of the Polypeptide Division Method into separate Domains [113]

Due to the compact nature of domains they should have a hydrophobic core, owing to the fact that hydrophobic amino acids attempt to escape their water based surroundings by burying themselves into the core of the protein [121]. Secondary structure boundaries also indicate the possible location of domain boundaries, as a domain boundary is unlikely to be found in the middle of an array of hydrogen bonds [131].

2.5 Protein Sequence Domain Databases and Classification:

It is also possible to identify some domains based on sequence alone, because of the evolutionary preserved functionality of a domain is contained within its sequence. A domain sequence can be conserved to maintain function through evolution but the sequence regions in between domains are likely to be more variable. The identification of domains based on sequence can be undertaken very quickly by sequence alignment programs such as BLAST, a program for pairwise sequence alignment. Searching these domain sequence databases can automatically, speedily and efficiently, identify domains based on Schultz *et al.* [110].

2.5.1 ProDom:

The ProDom database [112] looks for sequentially preserved stretches of a protein sequence, using the algorithm DOMAINER [115], which compares the protein sequence to every other solved protein sequence, and clusters homologous areas while looking for disagreement in domain borders (Figure 2.22) [14]. Protein sequence is a prerequisite for identifying structure.

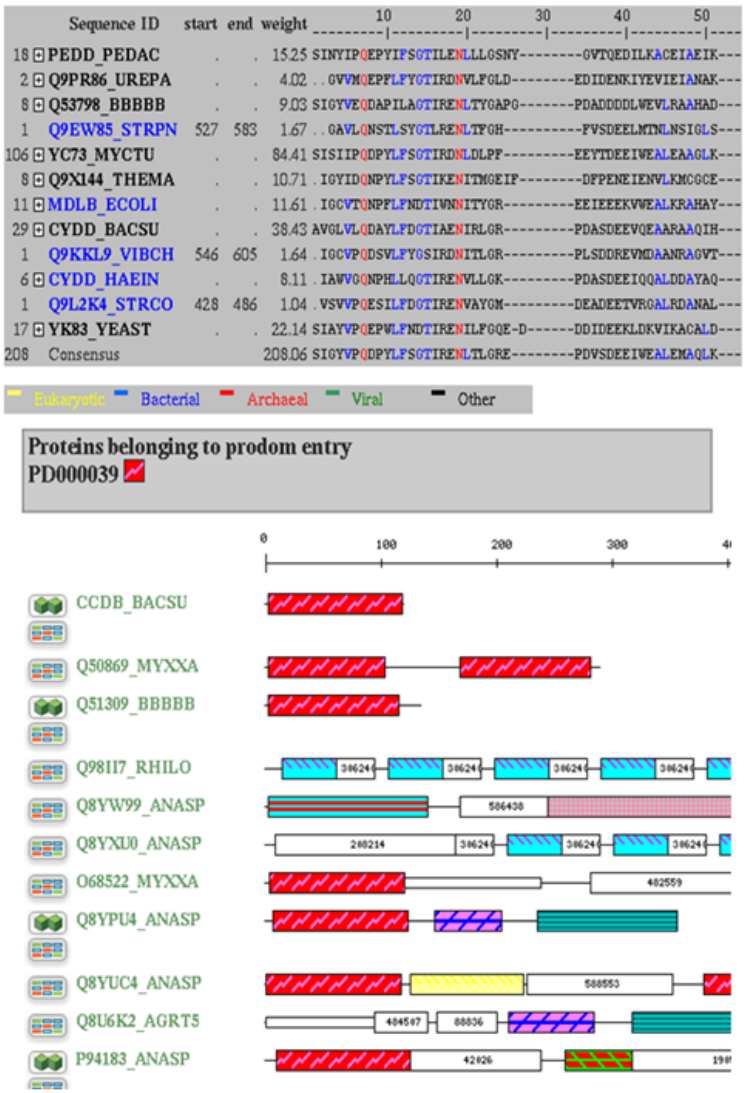

Fig. 2.22 ProDom Sequence comparison (with listing homologs) and domain identification

2.5.2 Pfam:

As the name would indicate, Pfam is primarily concerned with the family to which a protein belongs, but can also reveal information regarding domains. It uses multiple sequence alignments generated from hidden Markov models (HMM) [34] using the HMMER3 algorithm [35, 65]. HMMER3 is an accelerated version of HMMR which is fast enough to be able to carry out large scale genome or metagenome analyses [34]. PfamA is a subset

of Pfam which cross examines the Prodom database with domain information, and has now been incorporated into the iPfam domain database (Figure 2.23).

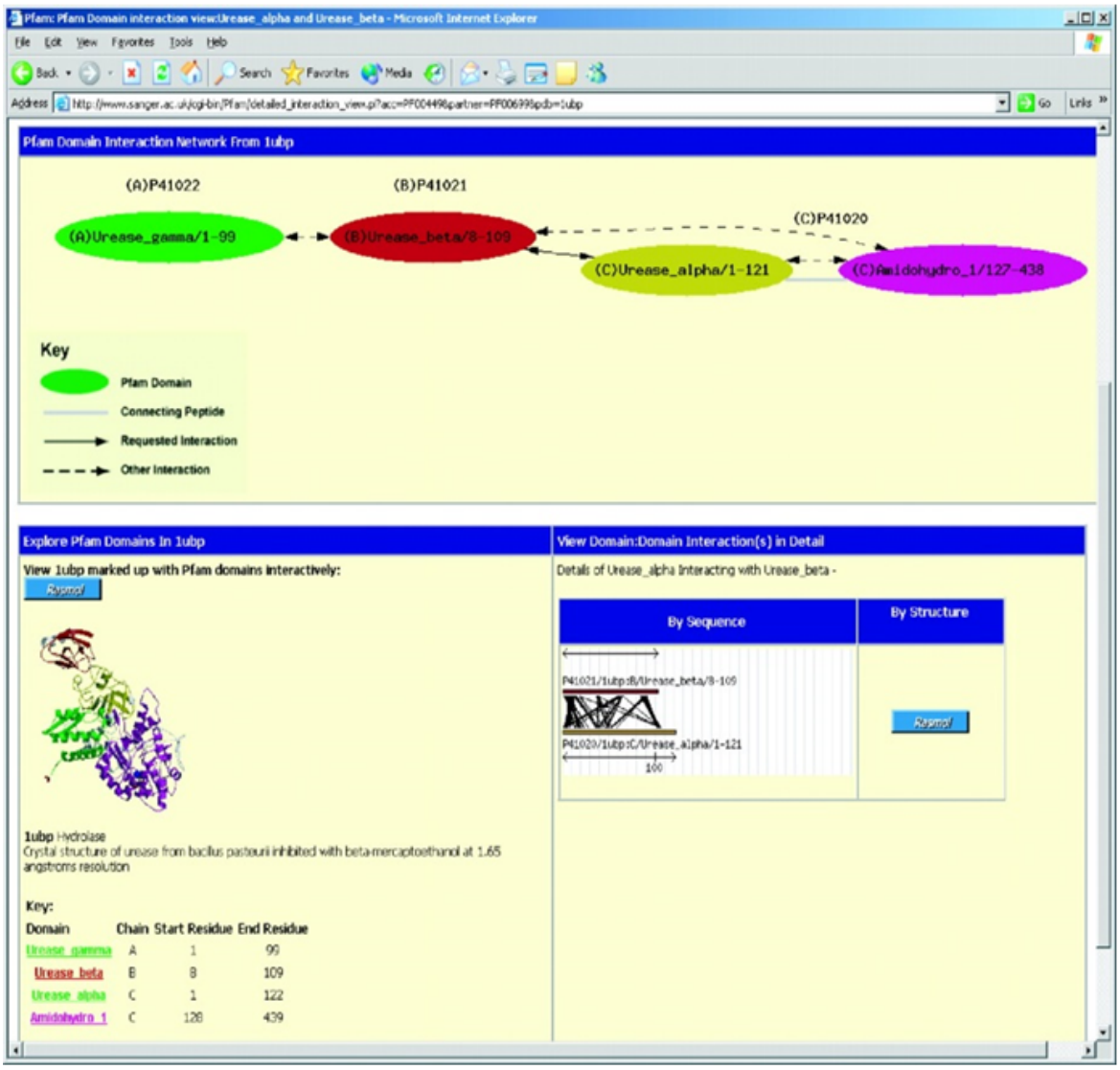

Fig. 2.23 The iPfam web page depicting the interaction between components of the urease complex in the PDB entry 1UBP [33]

2.6 Protein Structural Domain databases and Classification:

Databases that classify structures have domains as their fundamental unit of classification. SCOP (Structural Classification Of Proteins) [3, 76, 84], CATH (Class Architecture Topology and Homologous superfamilies") [89, 91, 92], FSSP (Fold classification based on Structure-Structure alignment) [57, 58] and 3Dee [24] all use different methods to identify domains in proteins (Figure 2.24). Often these databases employ a collection of different algorithms in combination to identify domains. These domain finding algorithms use methods referred to previously in the domain identification section [46].

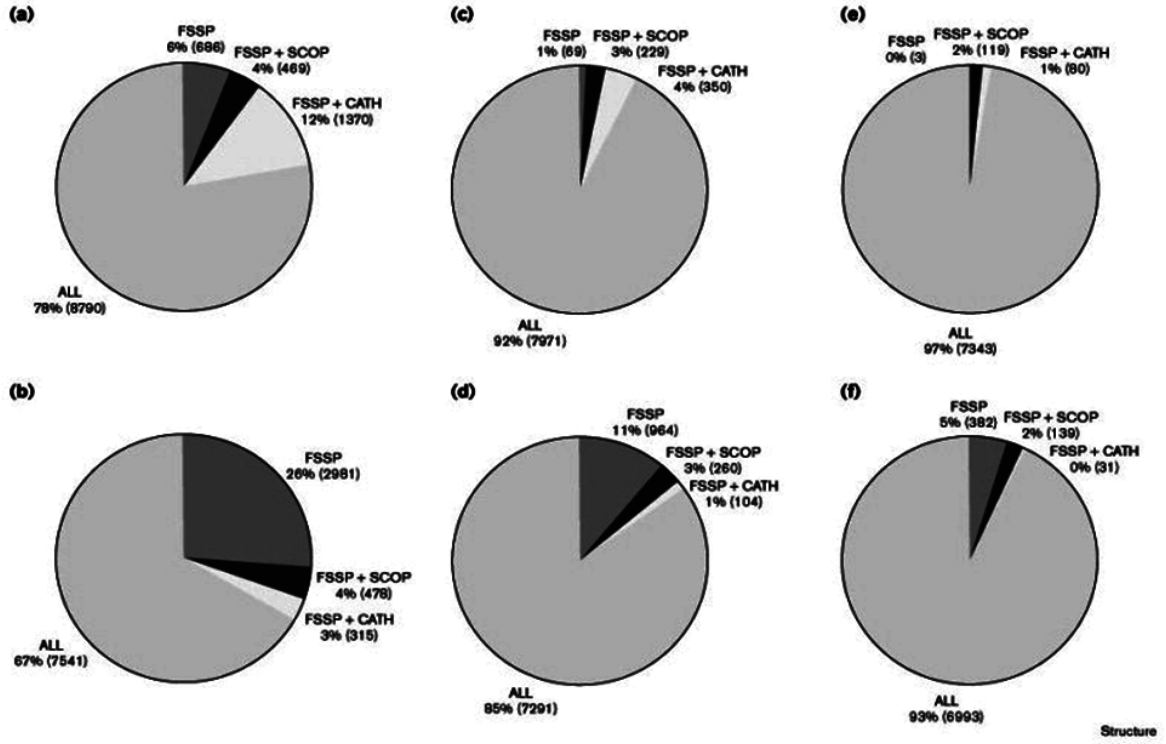

Fig. 2.24 Pie charts reflecting the agreement between pairwise matches in FSSP, CATH and SCOP (a, b) FSSP pairwise matches (Z-score $\geq$ 4.0) compared to CATH and SCOP matches at the fold and homology level, respectively. Numbers in parentheses indicate the number of pairwise matches in question. At this Z-score, agreement between the three databases is already high at both the fold and homology level. (c, d) Pairwise matches (Z-score $\geq$ 6.0) compared to CATH and SCOP as before. Agreement between the databases has increased by at least 15% at both the fold and homology levels. The difference between FSSP with SCOP and FSSP with CATH agreement has also reduced (e, f) Pairwise matches with Z-score $\geq$ 8.0. Already, agreement between the databases is as high as 97% at the fold level. Pairwise matches found in FSSP only are limited to three and the numbers of FSSP pairwise matches found in either SCOP or CATH (but not both) are very low [46]

2.6.1 SCOP:

SCOP examines proteins from an evolutionary perspective by analysing the preserved structural topographies, and in so doing creates relationships/ontologies amongst all known 3D structures. The database has a hierarchical structure. At the top of the hierarchy is the class, below that, the fold classification, followed by superfamily and family (Figure 2.25).

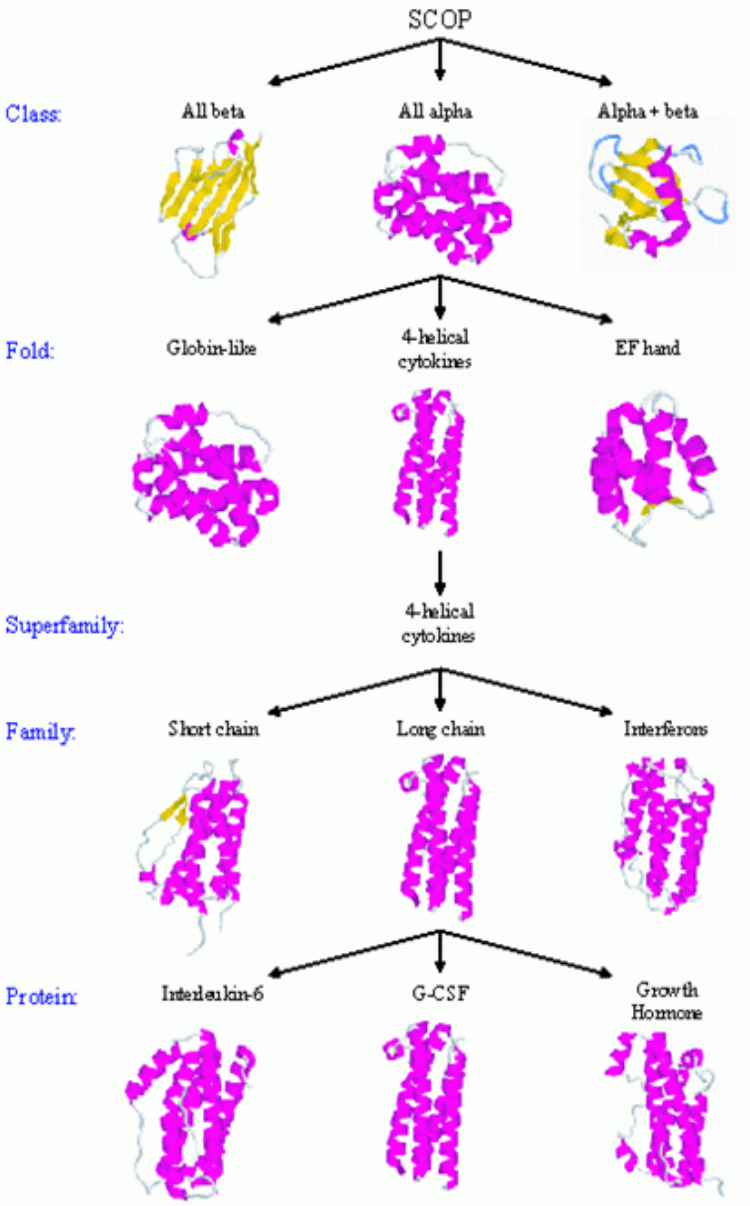

Fig. 2.25 SCOP algorithm intra-family relationships [53]

Based on secondary structure, a protein can firstly be assigned to one of the following classes: all α, all β, or a mixture of α and β (which can be broken down further into the ratio between α and β structures). Proteins in the same fold have their secondary structures organised in broadly similar arrangement. At the superfamily level proteins have much closer structural similarity whilst having low sequence identity. The strong structural similarity

suggests there is still a strong evolutionary link proteins within the same superfamily. Finally, structures are assigned to the same family if their sequence identity is greater than 30% [84].

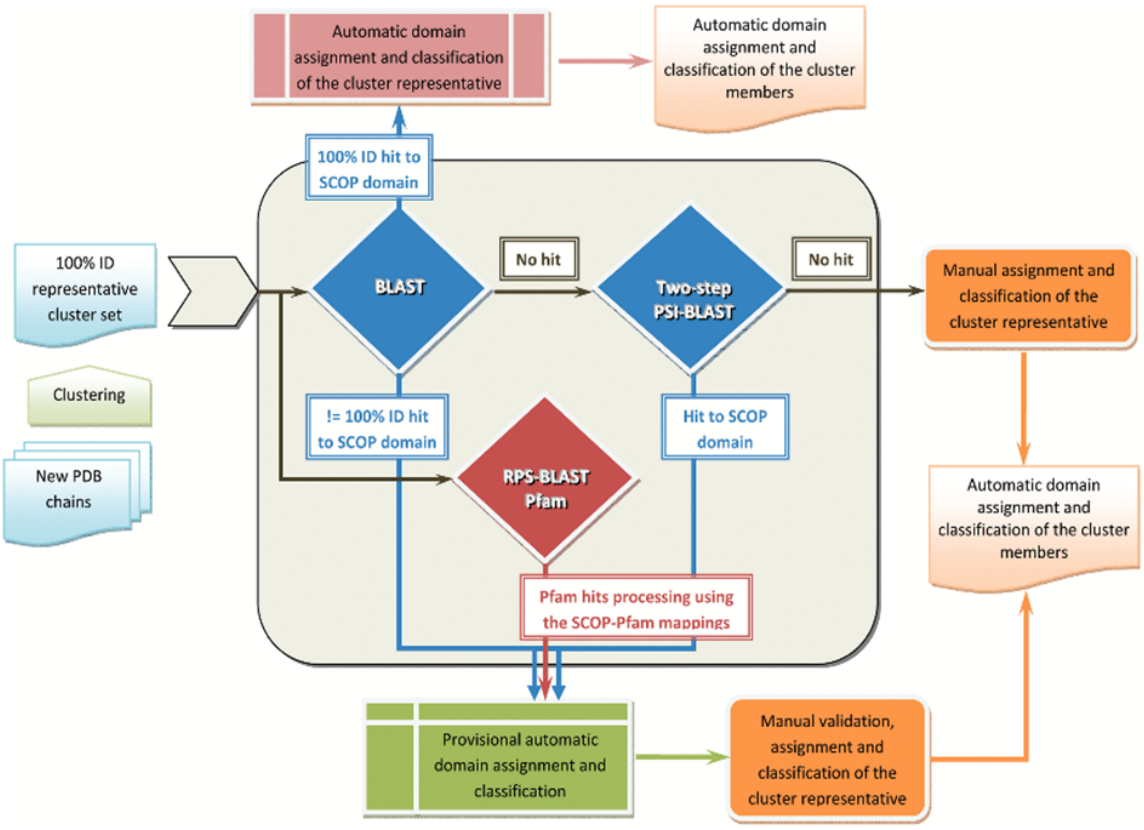

Fig. 2.26 Workflow of the SCOP update protocol. The update sequence set of new unclassified structures is derived from the PDB SEQRES record [4]

The recent advances in sequencing have allowed for a vast amount of sequence data and profile-based database searches to find new remote associations from sequence alone. For structure-guided searching of new evolutionary relationships at a Superfamily level, SCOP now integrates sequence to catalogue new structures and extends current categorisations (Figure 2.26). The advance of classifying protein families based on structure has enabled the detection of new evolutionary relationships in SCOP [4]. An extension of the existing protein superfamilies has provided both a structural and functional understanding of their fundamental members. Structural comparisons have exposed many instances of important

protein associations, signifying that substantial structural disparities exist within sequence families, which are more prevalent than previously thought.

2.6.2 CATH:

CATH partitions a protein into individual domains using a “committee” algorithm approach (where multiple algorithms combine to make a collective decision often referred to as “meta-analysis”).These algorithms are DOMAK [113], PUU [56] and DETECTIVE [121]. When these algorithms are unable to agree on a domain definition, the algorithm which provides the best score is chosen or additional information is acquired manually from the literature. DOMAK divides a protein indiscriminately into two parts and scores the division. Splits for which the score is large are those that have structurally distinct regions. The PUU algorithm examines inter-domain dynamics based on a harmonic approximation. The DETECTIVE algorithm examines intramolecular contacts and the presence of absence of a hydrophobic core within putative domains.

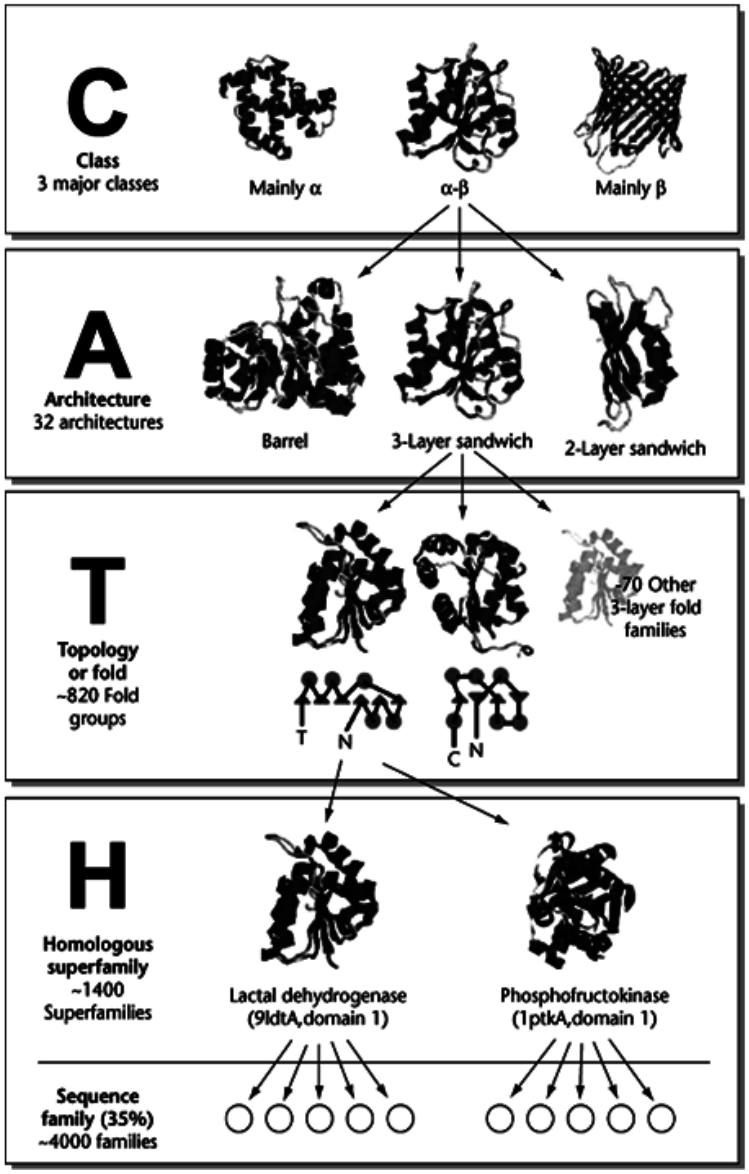

Fig. 2.27 CATH organisational system [138]

As with SCOP, the class of the domain is determined first, using the Michie *et al.* [79] method, which looks at secondary structure interactions, percentage of parallel and antiparallel β sheets and overall structural composition to assign the proteins to a category of mainly α, mainly β and a mixture of α and β. This class classification can be analysed further into collections of secondary structure to identify larger secondary structure subunits, which could be super secondary structure. As its name suggests CATH looks first at Class

of a protein, which is the total secondary structure content of the domain. Architecture is where there is strong structural similarity but no indication of homology, this is the same as a fold in SCOP. Topology (fold groups) is when a large-scale combination of topologies share specific structural features and finally 'Homologous Superfamilies' which are suggestive of an evident evolutionary connection, which also correspond to the superfamily level of SCOP. Fold recognition is a process for predicting the fold (path of the backbone) based on sequence. These folds are founded on overall shape and identify the connectivity of secondary structure, once joined together they can be connected by homologous superfamilies', where the domains can be investigated from an evolutionary preserved perspective, linking together overall structure and functionality (Figure 2.27). The CATH number each domain is given identifies the level of similarity at a fold level.

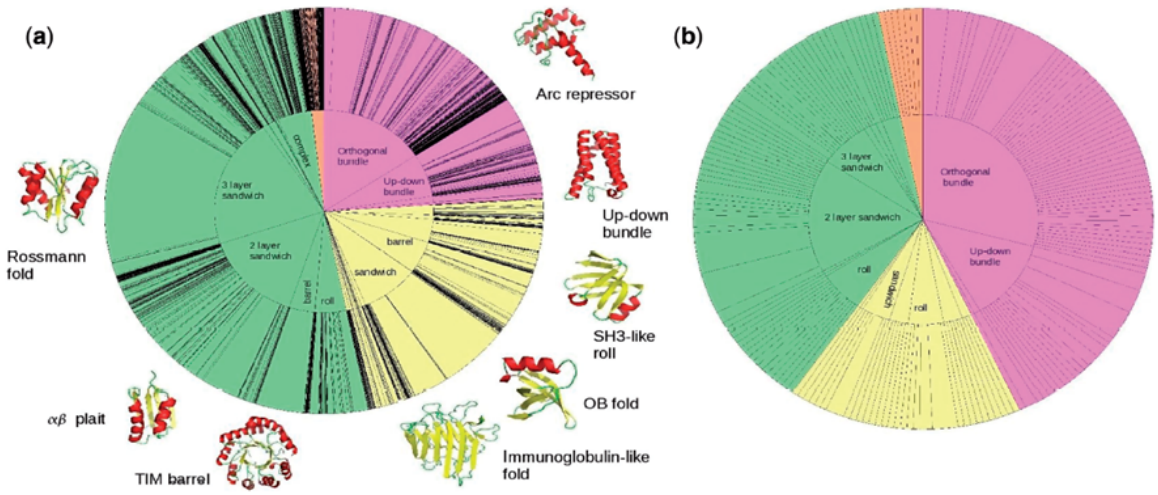

Fig. 2.28 'CATHerine wheels'. Segments are coloured according to class, namely pink (mainly α), yellow (mainly β), green ($\alpha\beta$) and brown (little secondary structure). The size of each of the segments represents the proportion of structures within any given architecture (inner circle) or fold group (outer circle). (a) The distribution of all non-homologous structures (2386) within CATH v3.3. Superfolds are represented as MOLSCRIPTS adjacent to the wheel. (b) The distribution of the 223 new non-homologous structures in CATH v3.3 [22]

CATH has now improved its search capabilities and grown to include 365 new superfamilies" (176 new fold groups), 29% of which came from the structural genomics (SG) initiatives

(30% fold groups) and 28% (22%) of which were membrane families (folds). Greater depth in the functional information for each CATH superfamily is now included through the incorporation of domain sequence lineages from Gene3D and by presenting their functional interpretations from different resources (e.g. GO, EC, Kegg and FunCat). Information is also now available on structural and functional differences across each superfamily and multiple structure alignments are available for clusters of close and distant structural families [22]. In addition possible evolutionary relationships derived from a multiple sequence alignments of enzyme superfamilies' are presented using phylogenetic trees which are further enhanced by functional features such as EC number and reaction mechanism (Figure 2.28).

2.6.3 FSSP/Dali:

FSSP also tries to find evolutionary connections between protein domains but does not ascribe fold families, superfamilies or classes in the process. It initially uses the PUU algorithm [56] to divide the protein into its principal domains. A more recent algorithm called "DomainParser2" [45] uses trained neural networks to enhance identification of domain partitions and is now in use. PDB90 [59] is a subset of the PDB and created by selecting the highest-quality structures from the PDB with maximum sequence identity of 90% ID. The Dali (Distance Matrix Alignment) algorithm [55] performs a pairwise structural alignment. Dali works by comparing intramolecular Cα-Cα distances between structures. Matching pairs of Cα's are unified with other matching pairs of Cα's to form larger matching structures (Figure 2.29).

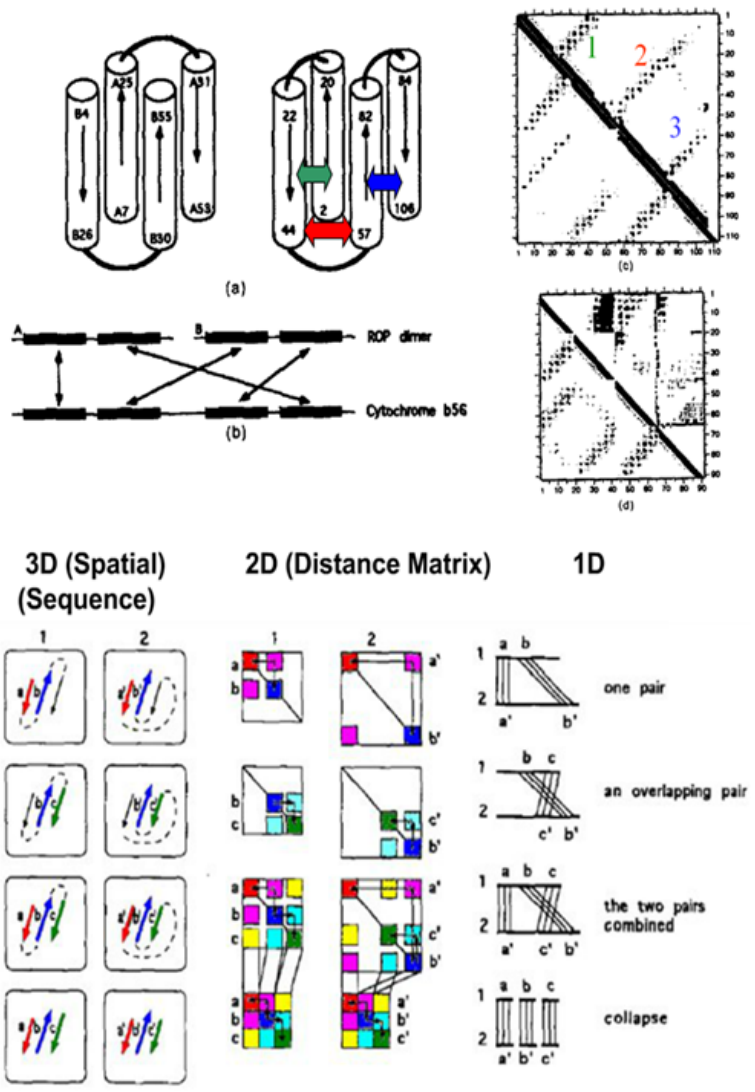

Fig. 2.29 3D to 1D conversion Dali method [64]

For each protein a pairwise structural match above a Z-score of 2 (calculated from the comparison between the representative and homologous sets) is made, which is further processed by a clustering algorithm, dividing the pairwise structural associations at Z-scores 2, 3, 4, 5,10 and 15 producing a cut-off index system.

2.6.4 HSSP:

HSSP (homology-derived structures of proteins) is a database that amalgamates structural (2D and 3D) and sequence information (1D) [105]. For each protein of known 3D structure, the database presents all sequence homologues, aligned to the protein of known structure [106]. Homologues are expected to have the same 3D structure as the protein to which they are aligned [109]. This means the database is not only composed of sequence aligned families, but also a database of inferred secondary and tertiary structures [108]. Probable secondary structure can be inferred from the protein of known structure to homologues [26]. Tertiary structure models can be constructed from the alignment with the known structure acting as a template (Figure 2.30) in a process known as protein homology modelling or protein comparative modelling.

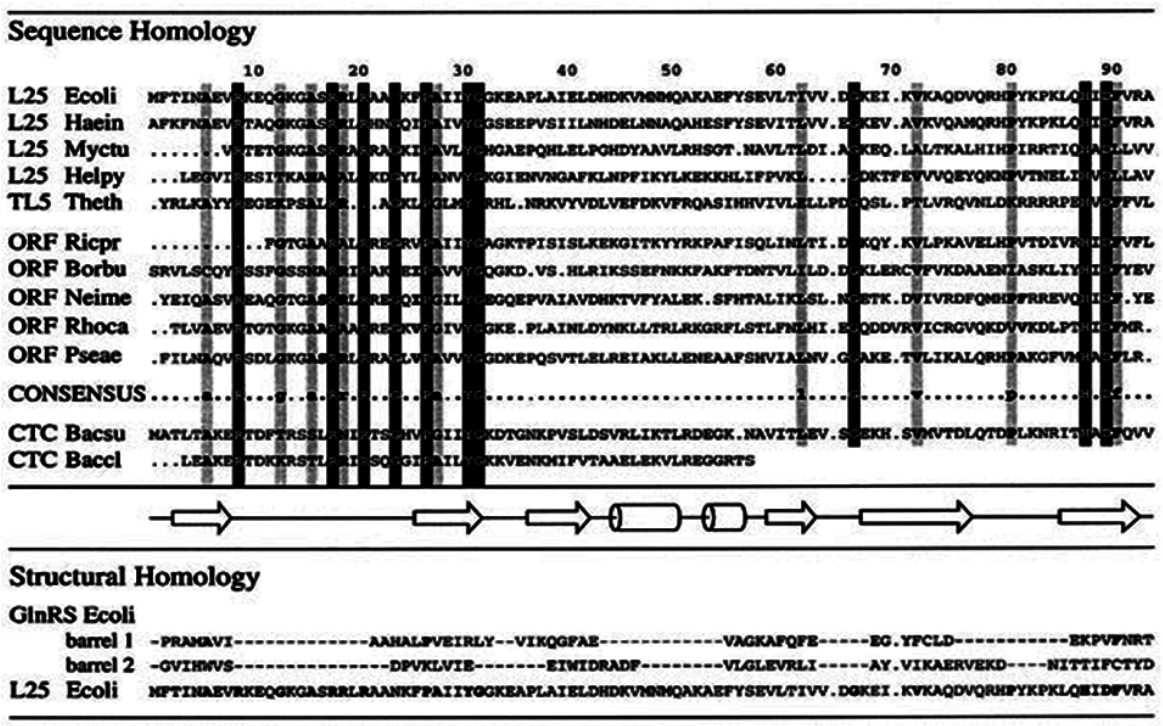

Fig. 2.30 sequence homology. Alignment and consensus sequence for protein and DNA-derived peptide sequences homologous to L25. Middle, cartoon of β-sheets (arrows) and α-helices (cylinders) found in the NMR structure of protein L25. Bottom, structural homology Alignment of the structurally analogous β-strand and α-helix sequences of the two β-barrels [119]

2.7 Protein Flexibility and Native State Dynamics:

Once the polypeptide chain has folded into its native structure it becomes functionally active. X-Ray Crystallography and NMR are able to solve the structure of a protein but give relatively little direct information on its dynamic behaviour.. Proteins are highly dynamic, and this plays a critical role in function. Proteins are capable of undergoing large domain movements, particularly in response to a change in environment or in interaction with another molecule e.g. in the case of an enzyme binding a substrate. Protein motions can occur at different levels of the structural hierarchy. These can be as small as inter atomic vibrations, flip-flopping of short loops, or rotations of amino acid side chains. On a larger scale domain motions occur and subunits shift. These are described as functional movements if they are coupled with ligand binding and enzymatic activity [93]. Structures solved with and without a substrate or substrate analogue bound, allow us to study the conformational change that occurs during function.

2.8 Protein Domain Movements Methodology:

Conformational change in proteins ranges from local motions, which include loop movements, side chain rearrangements and atomic fluctuations, to rigid body motions which include subunit and domain movements alongside more complex motions that occur in folding or unfolding. A protein domain movement is often the result of a functional necessity [23]. To establish whether a "dynamic domain movement" is present the structure of the protein must be solved by X-ray crystallography or NMR in different functional states [62]. These two structures or conformations can be evaluated computationally to investigate any dynamic domain movements present. It should, however, be noted that having only a small number of conformations available is limiting when considering the full extent of a protein's flexibility.

2.8.1 DynDom:

DynDom is able to analyse the conformational change between two structures of a protein [49] using rigid body kinematics (Figure 2.31). The underlying model of the methodology is one where the protein comprises quasi-rigid domains connected by interdomain bending regions.

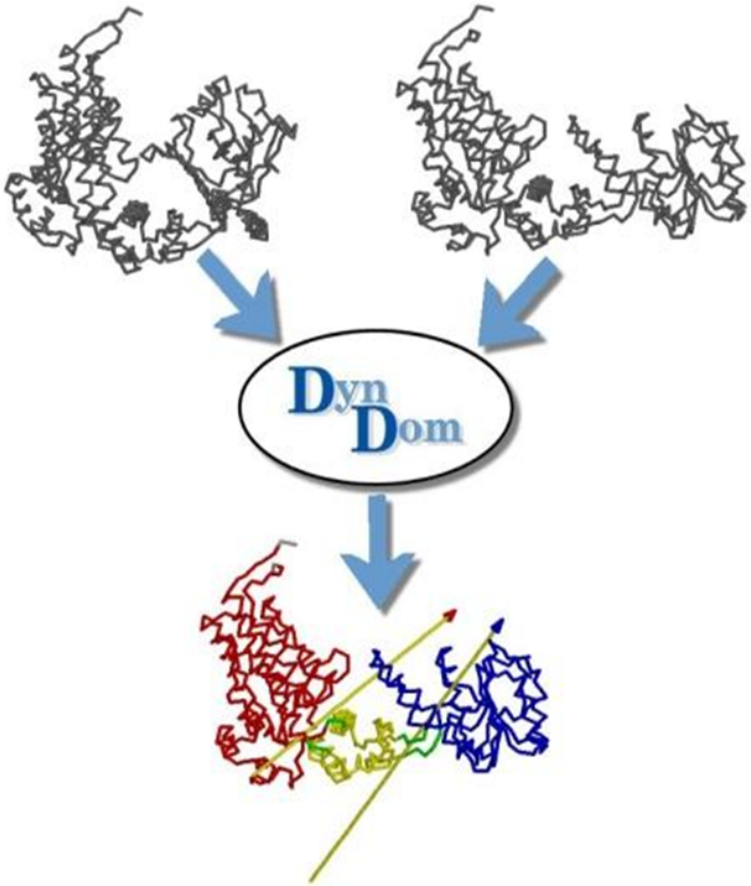

Fig. 2.31 Illustration of the DynDom process [97]

The advantage of DynDom is that its domain definition is based on movement of the quasi-rigid bodies not on structure alone. It is able to identify dynamic domains and the bending regions which link them (Figure 2.32).

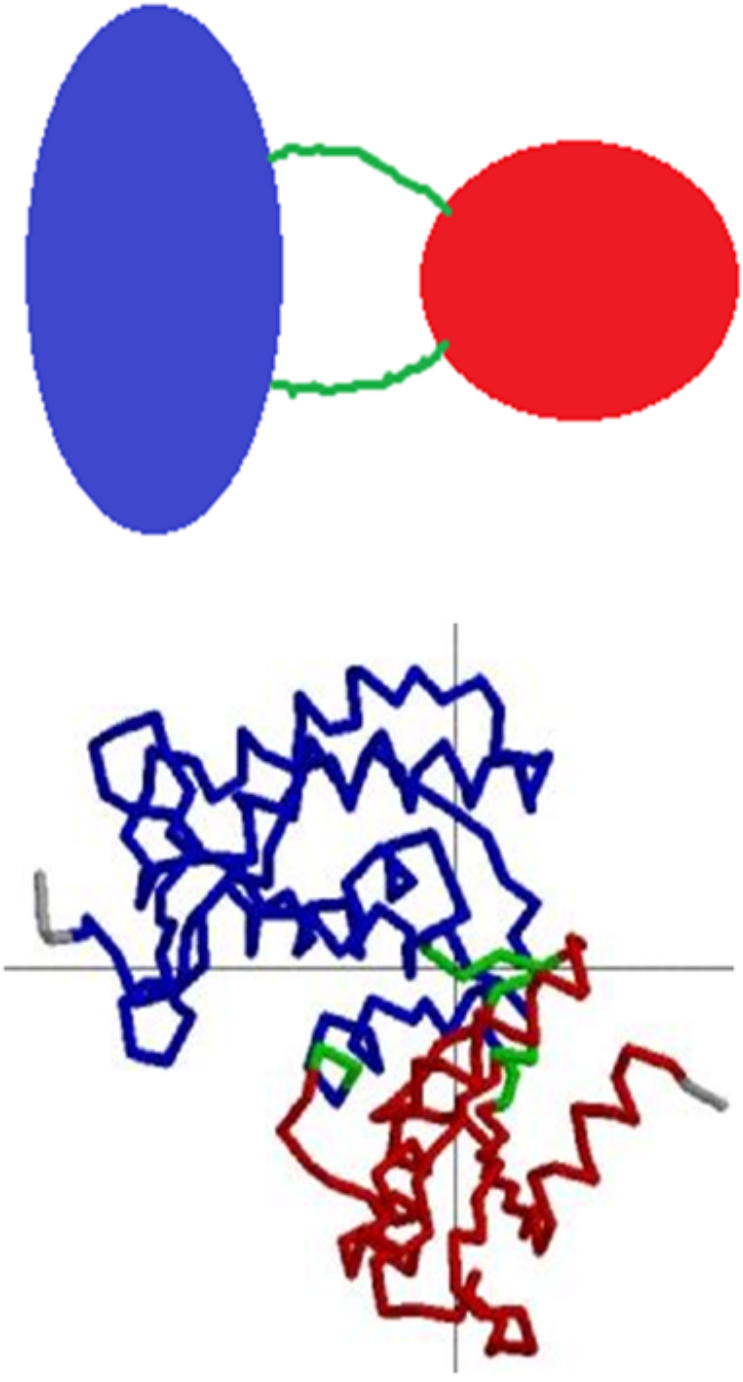

Fig. 2.32 Simplified Domain Model based on a DynDom structure (2O3S) (top) simplified domain representations (bottom) backbone highlighted: Domain A (Blue) Domain B (Red) Bending Region (Green) [97]

There are three stages in the characterisation of a domain movement. First the identi-

fication of dynamic domains, second the determination of the interdomain screw axis and finally the identification of the interdomain bending regions [48, 49, 51]. DynDom is part of the Collaborative Computational Project 4 (CCP4) a combined methodological resource for X-ray crystallographers. DynDom can be downloaded and run as a program. Alternatively it is run via a webserver. Results of previous runs are also available to browse online (Figure 2.33).

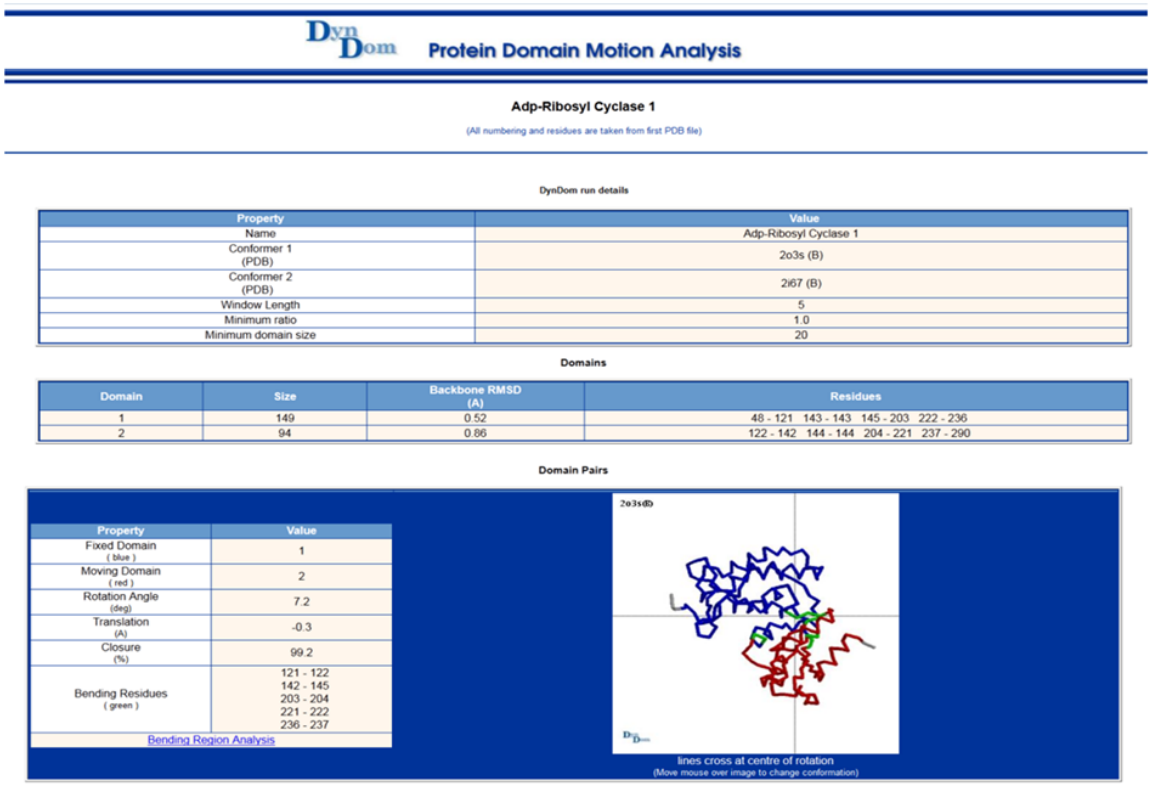

Fig. 2.33 An example of a DynDom entry in the DynDom online database [97]

Identification of Dynamic Domains:

To identify dynamic domains the two conformations are superimposed. Rotation vectors of backbone segments are calculated by analysing their rotational displacement between the two conformations. This is done segment-by-segment along the polypeptide backbone using a “sliding window” [50].

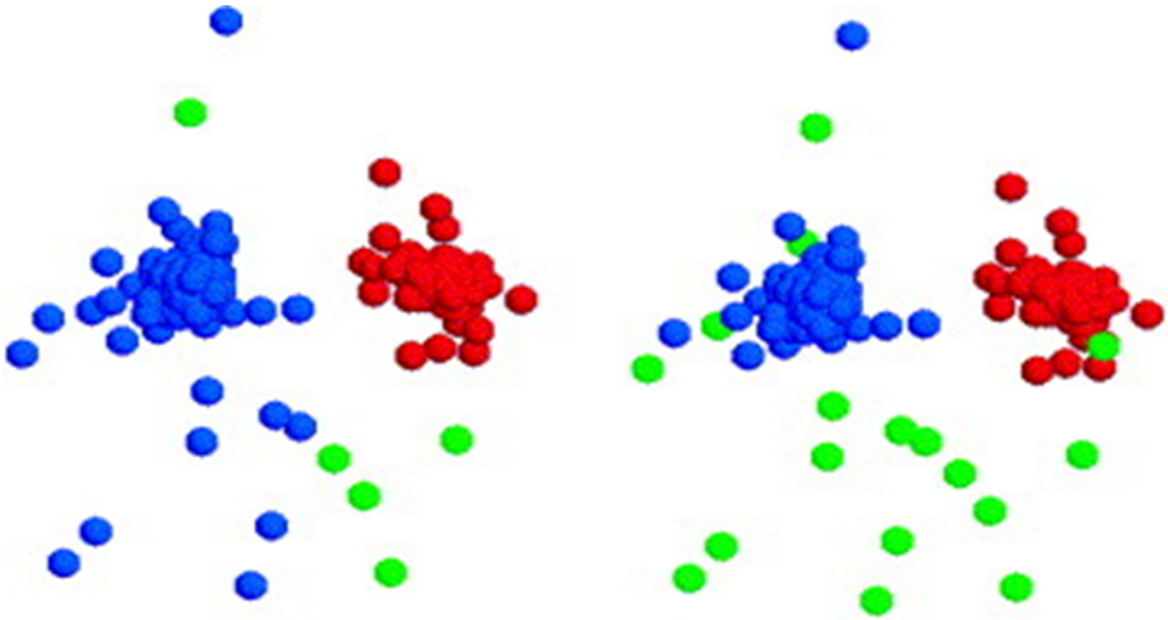

Fig. 2.34 Rotation points from the conformational change seen in 1IGT chains B and D. One domain (fixed) is coloured blue, the other red (moving) and bending residues green. In (left) from DynDom version 1.02, few residues are assigned as bending, in (right) using DynDom version 1.50, more are assigned as bending [51]

Treating the components of the rotation vectors as coordinates gives a set of points in 3D space. A K-means clustering algorithm (a non-hierarchical clustering algorithm) is then applied to these points to define clusters of rotation vectors, yielding potential dynamic domains (Figure 2.34). DynDom gives a result if displacements between domains are greater than displacements within domains. If the ratios of these two types of displacement are less than a threshold value specified by the user, this domain movement is not analysed [49]. The number of clusters is defined by K (starting from 1), meaning the whole protein is treated as one domain, and increases incrementally until a termination condition is met (see below). It is possible that a cluster of rotation vectors could correspond to two or more well separated (in space) regions in the protein, with each cluster rotating together. If the heavy atoms (Carbon, Nitrogen, Oxygen and Sulphur but not Hydrogen) belonging to a cluster are not associated by a network of distances of less than 4Å, the residues in the protein corresponding to the cluster will then be split into domains. Two important criteria that control running of the

program are:

- If a domain contains fewer residues than the minimum domain size set by the user, then segments from this domain are united with larger domains. If the sum of the domains from any single cluster are smaller than the minimum domain size, the process stops, unless this is the cluster found when k = 2.
- For every domain larger than the minimum size, the program checks which are connected through the backbone and calculates the domain displacement to intra-domain displacement as a ratio for every connected pair; if the ratio is less than the user-specified minimum (the second criteria) this pair is not analysed; otherwise it is.

The program finds the largest number of clusters for which all connected domain pairs satisfy both the minimum domain size criterion and the ratio criterion. These domain pairs are then analysed in terms of interdomain screw axes. If there are no connected domains to satisfy both criteria, DynDom will not run successfully. In the latest web version of DynDom, the window length is set initially to five residues but if there is a null result it increases in length in two residue increments until a domain motion is found, or the window length exceeds 15 residues [70].

Determination of the Interdomain screw axis:

In rigid body kinematics Chasles theorem states that, given a rigid body in two positions and orientations, the movement of the body can be represented by a screw movement about a unique axis [19, 41]. DynDom carries out this calculation and determines the position of the screw axis for the movement of all domain pairs for which there is a direct connection. The position of the screw axis in relation to the body of the protein in the two domains can reveal a great deal about the domain movement.

Determination of the Interdomain bending region:

After identifying the protein's domains and hinge axes, DynDom then determines the inter-domain bending regions. If one domain is labeled fixed and the other moving, there will be a rotational transition in the connecting region between the two domains. The bending residues are those that connect the domain pairs but also have rotations outside the main distributions for any cluster. The clusters are modelled as 3D normal distributions with residues in the bending regions being outside ellipsoids of constant probability with p=0.2. These are of importance as they play a crucial role in controlling domain movements. This domain decomposition is presented at the DynDom website [97] as a colour coded pairwise sequence alignment showing any substitutions, insertions and deletions between the two structures (Figure 2.35).

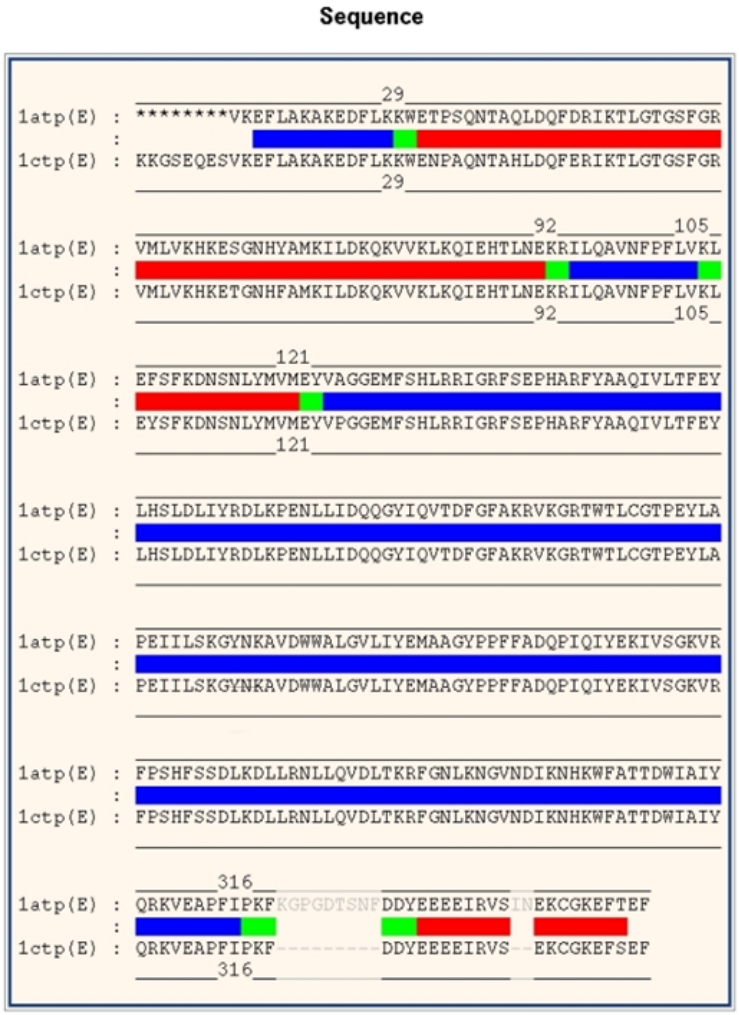

Fig. 2.35 Colour coded sequence alignment, Domain A (moving, red), Domain B (fixed, blue) and Interdomain Bending (Green) [97]

If the interdomain screw axis is close to a bending region (when the axis is within 5.5Å from any C-α atom of a bending region residue), the bending region is assigned as a "mechanical" hinge as it indicates that the rotation is being controlled by this bending region [49]. If one or more mechanical hinges exist in the interdomain bending region, this screw axis is referred to as an "effective hinge axis".

The information on DynDom regarding domain decomposition is presented at the website (1.9.3) and is also stored in a Rasmol script file (which can be downloaded from the DynDom

webpage; Rasmol is molecular viewing computer software [107]). The data file comes in two variants, the scrpt_IN file and the scrpt_PD file, although both ultimately produce the same result in Rasmol. The scrpt_PD file retains the original PDB numbering, which occasionally differs between the two structures. The scrpt_IN however synchronises the PDB numbering by starting them both at 1 and counting both incrementally; this means that the residue number in one structure directly corresponds to the residue of the same number in the other structure.

2.8.2 DynDom 3D:

DynDom3D is a program for the analysis of domain movements in large, multi-chain protein complexes. As with the original DynDom program, it can be used on any protein with two structures indicating a possible domain movement. Unlike DynDom, whose primary focus is on protein analysis, DynDom3D can be applied to any biomolecule [95].

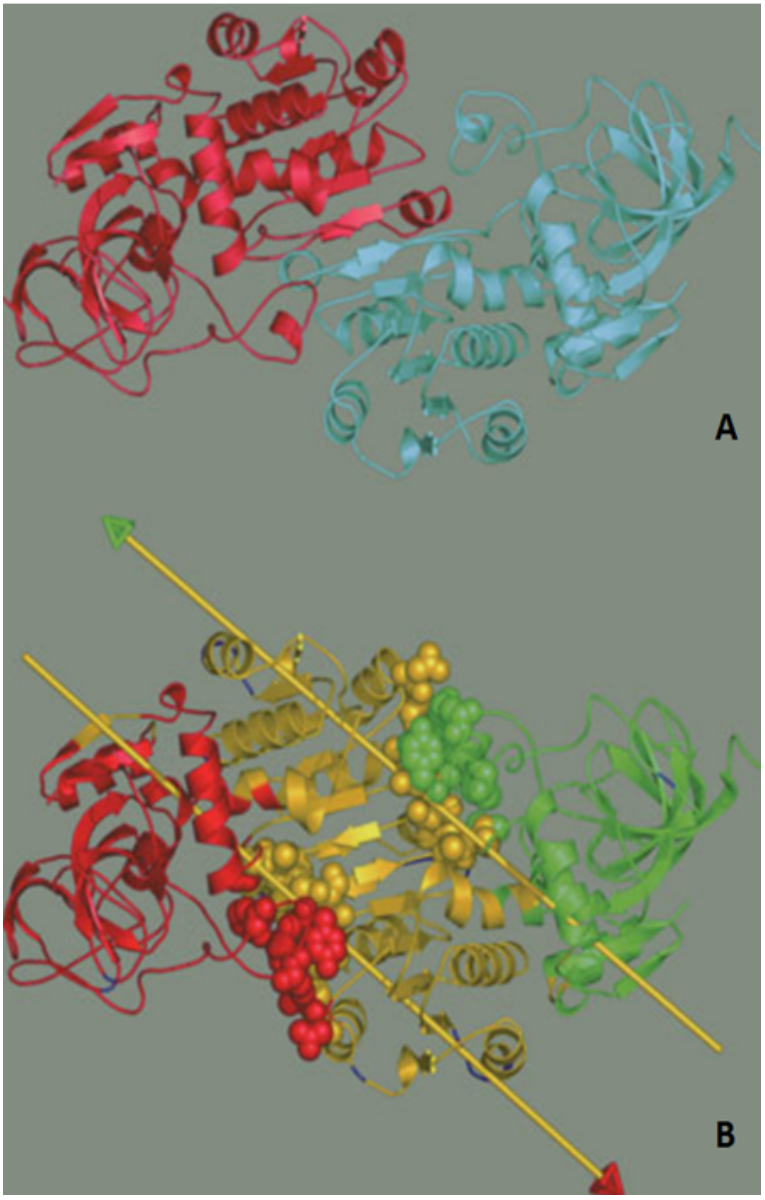

Fig. 2.36 Diagram of LADH: (A) Subunit colouring showing the two subunits (B) Dynamic domain colouring and interdomain screw axes. The two coenzyme binding domains form a single dynamic domain (yellow) at the center and the two catalytic domains form separate dynamic domains (red and green) [95]

The method overlays a cubic grid on the biomolecule and a block at least as large as each grid cube is placed at each grid point. The movement of atoms from the two structures within each block is analysed for its rotation vector. Clustering of rotation vectors indicate blocks that rotate together, indicating in turn collections of atoms rotating together forming

dynamic domains. The relative movement of domains is described by screw axes using Chasles theorem (Figure 2.36). DynDom 3D uses five main parameters: the number of atoms for the minimum domain size, the ratio between the interdomain and intradomain displacement, a grid size, a block factor, and a block occupancy percentage [95].

2.8.3 Rigid Domain Method:

The method presented by Nichols et al, requires two conformations of the same protein and determines rigid domains [87]. A rigid domain is determined by measuring the distance between two residues and checking whether it is the same from one confirmation to another. These residues need not be spatially or sequentially close. The number of amino acids allowed in a domain is specified as a parameter. This method identifies rigid domains using a displacement parameter, which can be calculated with a distance difference matrix for all residue pairs followed by a comprehensive search for rigid amino acid triplets (triplets being the smallest initial size but increases in increments, if necessary). The disadvantage of this method is that it is CPU demanding rendering it unviable for large structures. The method is not extended to determine axes or hinge bending residues.

2.8.4 HingeFind:

Like DynDom, HingeFind [63, 136, 137] identifies and characterizes domain movements using rotation axes, but HingeFind looks for pure rotation axes rather than screw axes. HingeFind superimposes two structures to determine the degree of deformation. Structures are superimposed using the least-squares best-fit method using sequence alignment to identify equivalent amino acids. An “adaptive selection” algorithm, a variation of Lesk’s sieve fit algorithm [71], selectively matches residues from a domain in comparison to a reference domain by minimising the root mean square deviation and adding new atoms according to a tolerance value. Once subsets of atoms form a rigid domain, they are excluded from

further analysis. When the protein is completely segregated into domains the HingeFind program stops. The program proceeds by characterizing domain movements between the rigid domains using effective hinge axes, as the movement of two rigid bodies is controlled by an interdomain bending region joining the protein domains (Figure 2.37).

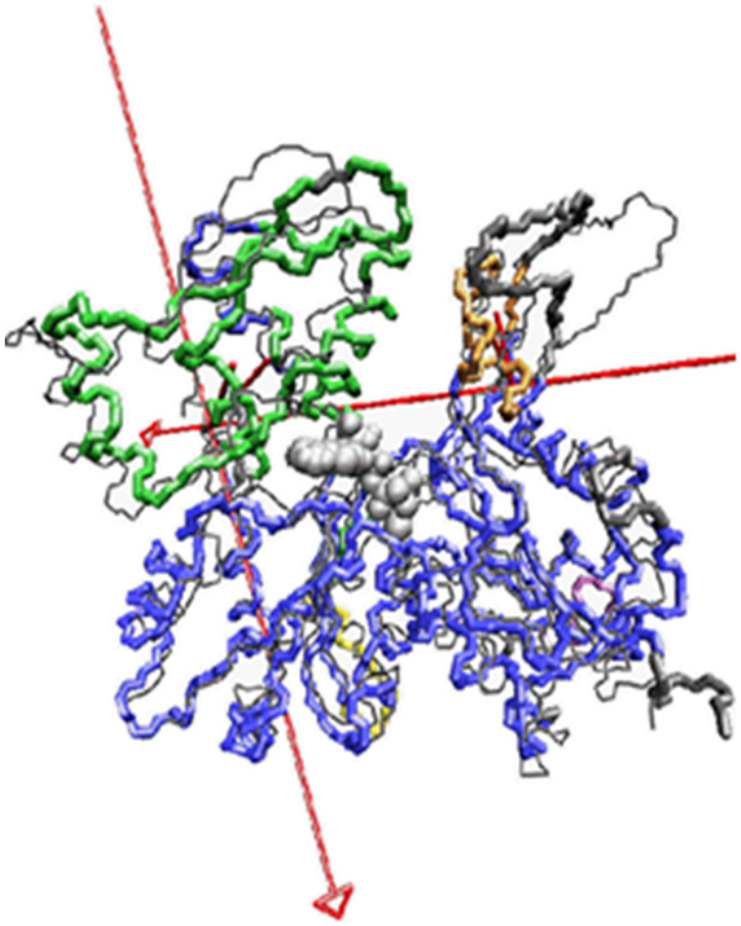

Fig. 2.37 Backbone trace of F-actin structure. The nucleotide and divalent cation of the comparison structure are rendered as grey van-der-Waals spheres. The colour of the tubes codes for the partitioned segments found: reference-segment 1 (blue), segment 2 (green), segment 3 (orange), segment 4 (yellow), segment 5 (purple), no segment assigned (grey). The two structures are superimposed by a least-squares fit of segment 1. For segment 2 and segment 3, the rotation axis and "pivot" connecting lines of movements relative to segment 1 are shown as red tubes. The arrow indicates a right-handed rotation. The domain movements yield a closure of the nucleotide binding cleft. Segments 4 and 5 comprise only few residues [137].

2.8.5 DomainFinder:

The DomainFinder program, as with the DynDom and HingeFind, identifies and describes dynamic domain movements. Another similarity to DynDom is that it examines the polypeptide backbone, investigating the comparative positions in each conformation and the changes which describe the motion. It also analyses clusters of points in a parameter space. DomainFinder calculates deformation energy for each Cα, calculated from changes in distance from neighboring atoms. The program uses two constraints entered by the user; a deformation threshold (which specifies adequate rigidity), and a similarity threshold (which analyses the degree to which regions can be considered analogous and grouped together to form a single dynamic domain). A low value signifies comparatively rigid regions (suggesting probable domain regions) and high deformation energies suggesting more flexible regions.

The more rigid regions can be unified to form potential dynamic domains. Perhaps the biggest advantage for DomainFinder is that, it can also be used when only one structure is available. In order to do this a simplified variant of normal mode analysis is used [54] which can often produce clearer domain boundaries. It also delivers a useful measure of rigidity, permitting easier categorization of quasi-rigid domains, providing detailed descriptions of low-frequency protein motions with the decomposition of the protein into highly flexible regions, rigid domains and semi flexible transition regions.

Gerstein Methods:

Mark Gerstein is a Professor of Bioinformatics at Yale University and leads a group that has contributed a great deal to macromolecular motion research and has devised several methods for studying Protein Dynamics.

2.8.6 FlexOracle:

FlexOracle addresses the problem of locating the primary hinge site for hinge bending proteins. It is a program for predicting the location of hinge sites and might be compared with methods that predict domains using a single structure. It does this by calculating energetic interactions. Fragments are produced by cleaving the protein at a hinge site to produce possibly independently stable regions. The program is employed within the Database of Macromolecular Motions (1.9.2). For each structure, fragment pairs are created based on examining all potential cleavage sites on the polypeptide chain, calculating the energy of the fragments in comparison to the whole protein, and predicting the location of hinges where this measure is smallest [36]. The method is efficient because only fragment pairs produced by cutting at a single site on the protein chain are considered. There are three applications in this method; firstly a molecular mechanics force field is used to compute the energies of the two fragments cut at a single location. Secondly fragments are created in an identical fashion but with their free energies being computed using a knowledge-based force field, and thirdly fragment pairs are produced by cleaving at two points on the polypeptide chain and then calculating their free energies (Figure 2.38).

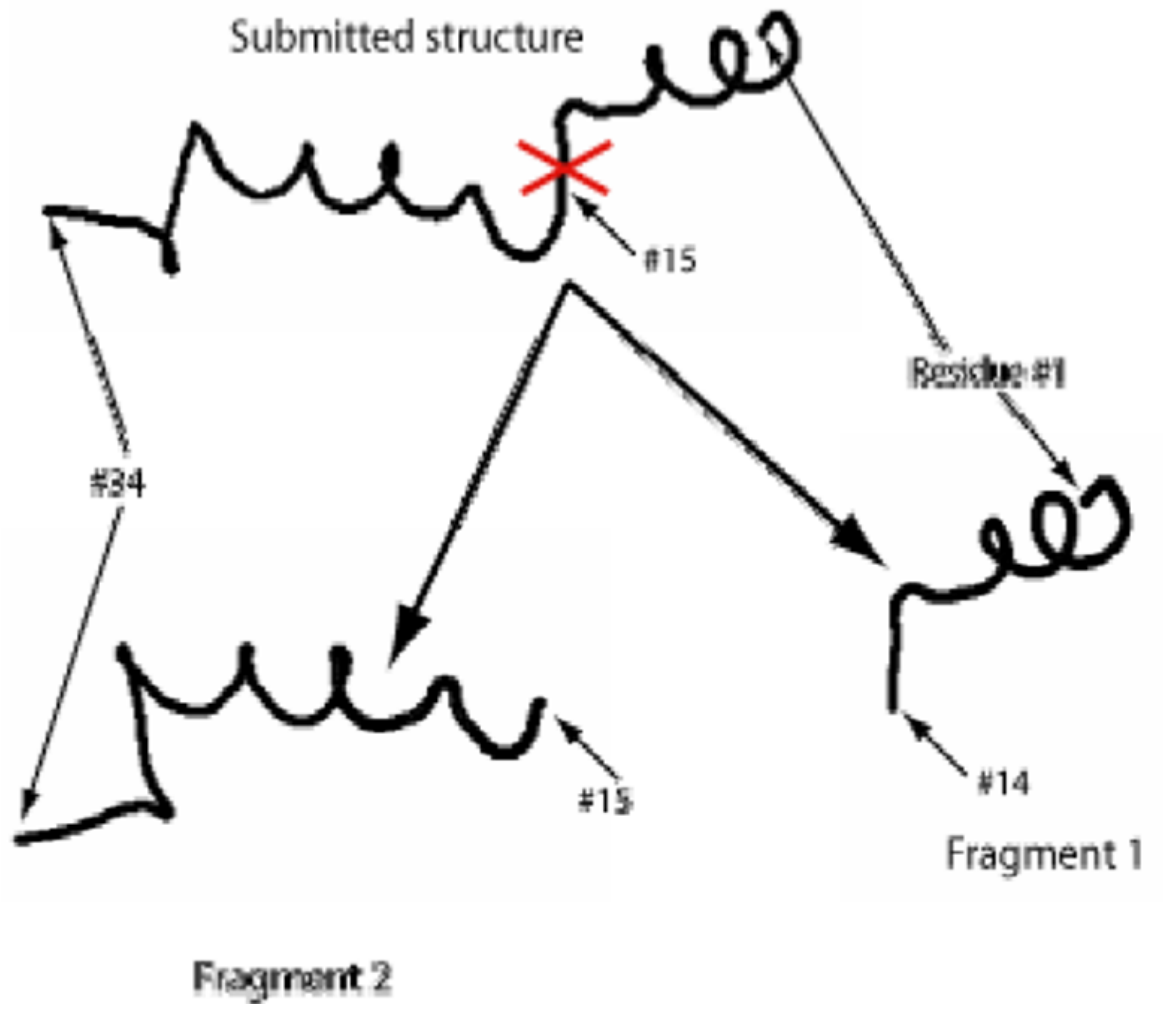

Fig. 2.38 A key step in the FlexOracle method: separating the protein into two fragments [36]

FlexOracle has been able to predict the hinge site for domain proteins, giving comparable results for the location of hinge sites as determined in the analysis of apo (no ligand) and ligand bound structures, but only when the ligand is a small molecule. Hinges often correspond to minima of the single-cut FlexOracle energy, but in the case of two-domain proteins encompassing one adjoining and one discontinuous domain, the hinge can instead exist close to the borderline between a broad high energy (equivalent to the adjoining domain) and wide low energy (corresponding to the unjoined domain).

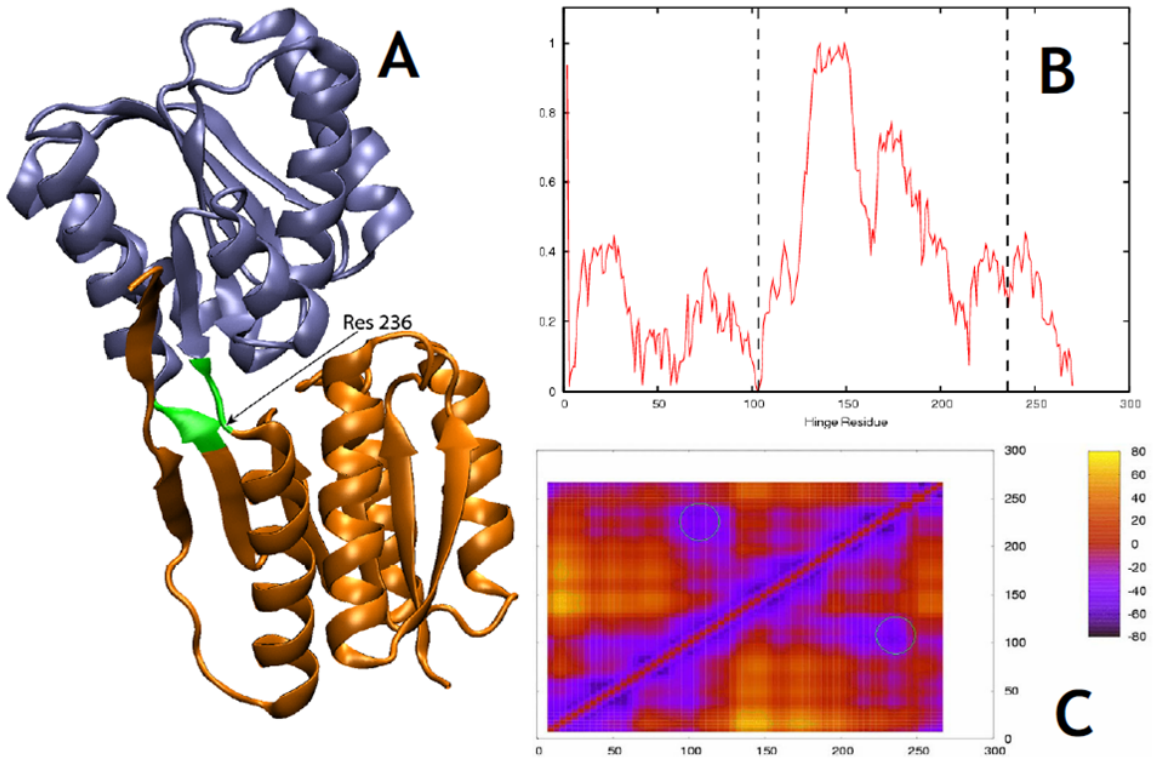

Fig. 2.39 (A) Ribose binding protein (IDRJ) (open). (B) The single-cut predictors suggest the hinge at residue 103, but less clearly at residue 235, false positives can be seen, at residue 135 and around residue 50 (C) The 2-cut predictor gave the correct result, as seen by the minimum faintly circled [36]

If the linker has within it narrowly spaced parallel strands, the hinge is inclined to occur a few residues into the high energy side of this boundary. With the exception of bound metal heme components, the two-cut predictor has been shown to work very well, being more accurate than the single-cut predictor (Figure 2.39) and effective for single and double stranded hinges, but not for triple stranded hinges.

2.8.7 RigidFinder:

Advances in structure determination can now solve large macromolecular complexes. New methodological approaches are required to deal with them [1]. The fundamental element of motion study has been the identification of moving rigid blocks by comparing different crystal

structure conformations, but current methods do not permit reliable block identification in very large protein structures.

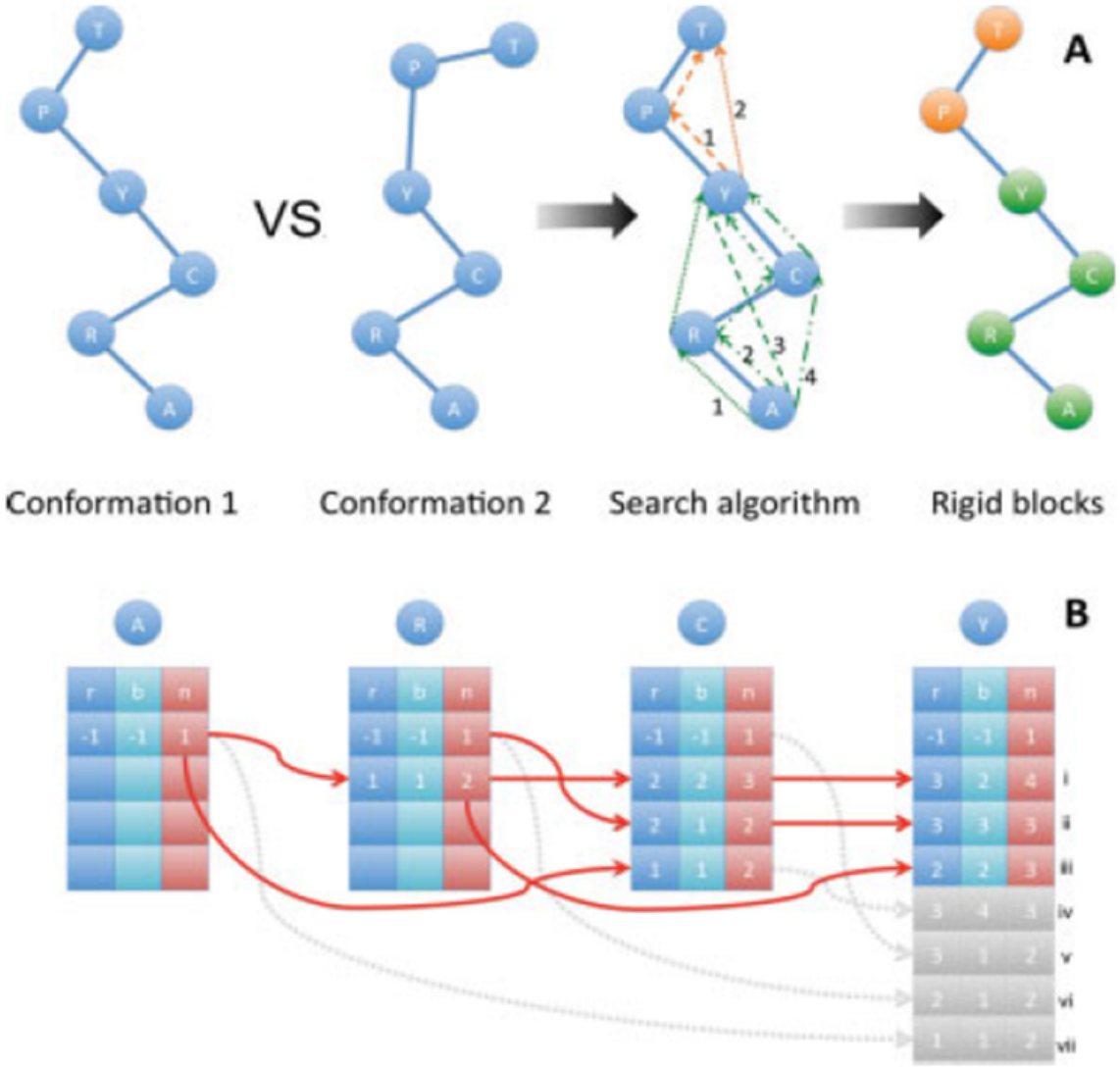

Fig. 2.40 RigidFinder method: (A) Example of all paths for block extension for the comparison of two protein conformations. The largest rigid block will consist of the first four residues (shown in green) and another rigid block will consist of the remaining two residues (shown in orange). (B) Quasidynamic programming to find the largest rigid block for the first four residues. Each block is tracked by: r = index of the previous residues in the block, b = index of the block as it is for the previous residue and n = number of residues in the block [1]

RigidFinder is a computer program which can identify rigid blocks from different conformations, from large complexes to small loops. RigidFinder describes rigidity in terms

of blocks, where inter-residue distances are preserved across conformations (Figure 2.40). Distance conservation, unlike the averaged values (e.g., RMSD) used by many other methods, allows for selective identification of motions. It is capable of finding blocks comprising non-consecutive fragments in oligomeric complexes. It uses quasi-dynamic programming search algorithm, which is fast on very large structures. It can be run at a web server (http://rigidfinder.molmovdb.org) providing illustrations at different scales such as loop closure, domain motions, partial refolding, and subunit shifts. The results of RidgeFinder give agreement with results of other methods.

2.9 Databases of protein domain movements:

2.9.1 Protein Structural Change Database (PSCDB):

The PSCDB project involved examining multiple entries of proteins determined with and without a ligand molecule bound, providing significant information for understanding structural changes associated to protein function [2]. 839 structural pairs in ligand-free and ligand-bound states (of monomeric or homodimeric proteins) were analysed. The aim was to characterise the motion coupled with ligand binding. The analysis yielded seven classes (Figure 2.41).

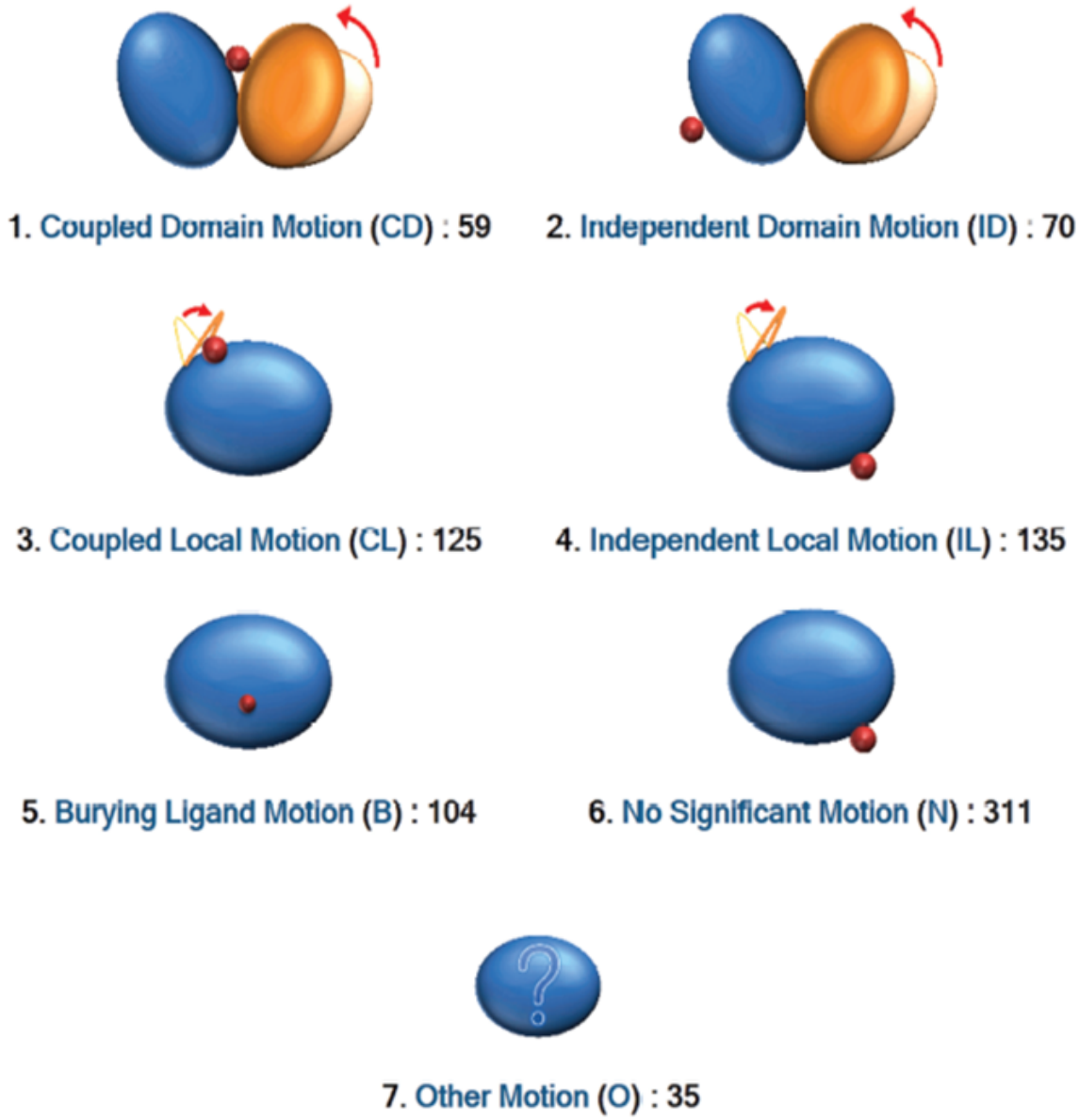

Fig. 2.41 The seven domain movement classes. The names, abbreviations, numbers and schematic figures of protein structural changes of the seven classes [2]

2.9.2 Database of Macromolecular Movements:

The Database of Macromolecular Movements [39] characterizes all two domain proteins into two main categories (Figure 2.42) predominantly consisting of “hinge” (where two domains travel towards one another in a perpendicular fashion) and “ predominantly shear” (where two

domains in close proximity slide over one another) [27, 40] http://www.molmovdb.org/cgi-bin/browse.cgi

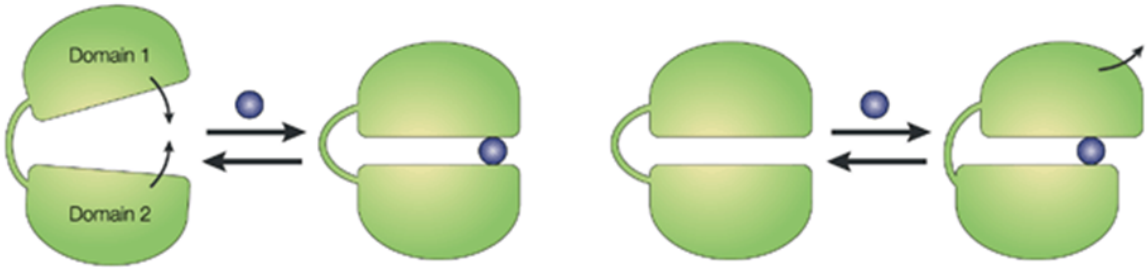

Fig. 2.42 Examples of the two domain movements suggested by Gerstein et al [40] (left) hinge (right) shear [124]

A shear motion is assigned when the protein movement maintains a well packed interface. Movements tend to be very small but they could combine to give a greater overall domain motion. When there is no sustained interface between the domains with relatively unconstrained packing, with the domains held together by a flexible linker region, a hinge motion is assigned. Large torsion angle changes in the linker regions produce a large domain movement. This often occurs with an axis passing through the interdomain bending region. The database of macromolecular movements currently has 37 domain movements classified as "predominantly shear" and 75 domain movements classified as "predominantly hinge".

2.9.3 DynDom Database:

The DynDom database stores the results of DynDom runs [70]. This online database allows users to upload their own PDB files or select them from the PDB. If the DynDom run is successful and use structures in the PDB the results will be stored in the "User Created Database" (UCDB). The UCDB originally started with just 24 proteins [48] but currently stores over 3000 examples. The "Non Redundant Database" (NRDB) is the result of an exhaustive analysis of all available protein structures to build a comprehensive database of all known protein domain movements. It comprises 2035 unique domain movements, within

1578 families [97]. This involved grouping proteins into families, clustering these to remove conformational redundancy and then using a Gram-Schmidt technique to select the best archetypal movements in each family, before finally running the DynDom program on these pairs [97]. Of the 2035 cases, 1822 are two domain proteins, which are abbreviated here to NRDB2d.

In a further development ligands in the NRDB have been cross-referenced with ligands found in the KEGG-LIGAND database [43] in order to identify functional ligands in enzymes. By finding ligands contacting the protein in one conformation, but not the other it was possible to identify ligands that might trigger the domain movement. Using this approach the non-redundant database was distilled down to a set of 203 enzymes where a domain movement is elicited by ligand binding. This gives dynamic information, including regions forming dynamic domains, hinge bending residues, and the hinge axes, together with ligand binding data. Within this set "Spanning Trigger-Ligands" are identified (139 examples) where "spanning" means both domains are in contact with the ligand and "trigger" means a ligand is present in one conformation, but not the other [96]. Unlike the NRDB2d it does not encompass domain movements affected by miscellaneous changes in external circumstances, not related to function. This data has been integrated within the DynDom relational database system and can be extracted using SQL [70, 96, 97].

Chapter 3

Methodology

3.1 Set Theory:

3.1.1 Definition of Sets

Set theory examines sets, associations and collections by mathematical logic. This is applied to objects/data with direct significance to mathematics, as the theory can define all mathematical articles [15]. A set can be regarded as a group of objects; when they are within a set, these objects are called "elements". Convention dictates that uppercase nomenclature denotes a set while lowercase nomenclature denotes the elements within the set [120]. Conventionally, when referred to, items in a set are defined in curly brackets ({}):

$$A = \{a, b, c, d, e, f, g\} \tag{3.1}$$

If an element belongs to a set then "$\in$ " is used. For example:

$$a \in A \tag{3.2}$$

When an element does not belong to a set "$\notin$" is used. For example:

$$i \notin A \tag{3.3}$$

There are two special sets which have their own unique notation. The universal set encompasses all elements regardless of other sets to which they might belong and is denoted by "U" which means "universal". The null/empty set where there are no elements present at all, is denoted by $\emptyset$ or $\{\}$ [74]. A set comprising only elements from a set is known as a subset. Subsets of a set are denoted by $\subset \setminus \subseteq$. It is possible to reverse the direction and have a superset with $\supset$ / $\supseteq$ (Figure 3.1):

$$\begin{aligned} A &\subset B \\ B &\supset A \end{aligned} \tag{3.4}$$

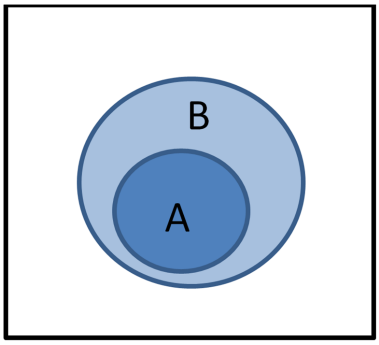

Fig. 3.1 Venn diagram representing subset A of set B

A proper subset is defined as being a subset which does not share exactly the same elements as its superset. In other words, if B is a proper subset of A, then all elements of B are in A but A contains at least one element that is not in B. For example, if $A = \{1,3,5\}$ and $B = \{1,5\}$ then B is a proper subset of $A : B \subset A$. If set $C = \{1,3,5\}$ then C is a subset of A, but it is not a proper subset of A because C is the same as $A : C \subseteq A$, $C \not\subset A$.

If set $D = \{1,4\}$ this would not be a subset of A, because 4 is not an element of A.

The union operator in set theory identifies all elements that belong to all sets defined and is denoted with the $\cup$ character (Figure 3.2):

$$A \cup B$$
$$A \cup B \cup C \tag{3.5}$$

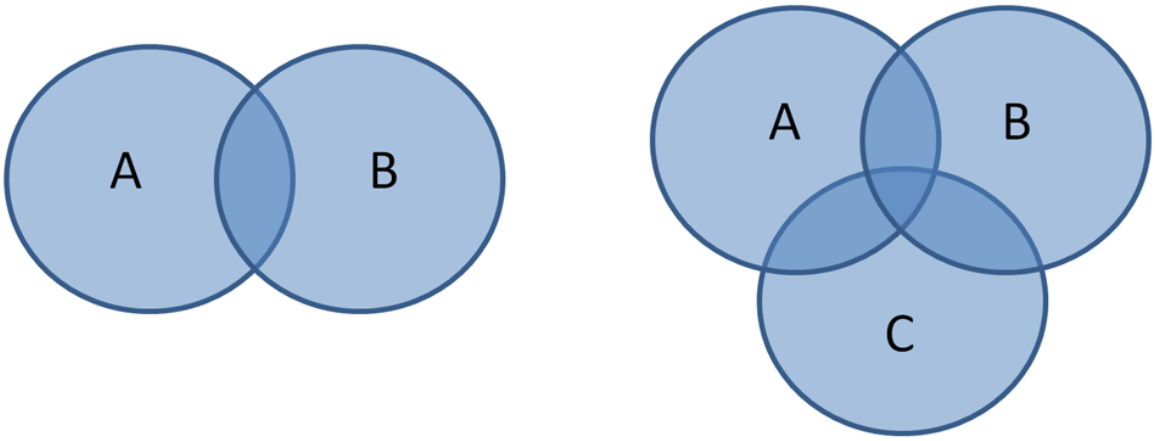

Fig. 3.2 Venn diagram representing union of sets

3.1.2 Intersection

Intersection in set theory defines only the elements which are shared by two or more sets, given by the $\cap$ symbol (Figure 3.3):

$$A \cap B$$
$$A \cap B \cap C \tag{3.6}$$

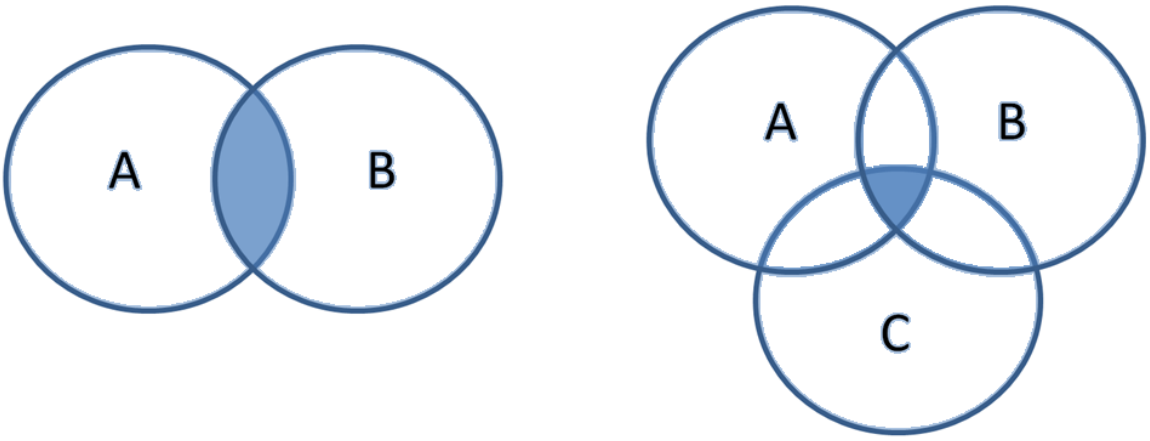

Fig. 3.3 Venn diagram representing intersection of sets

3.1.3 Set Difference

The set difference of B and A (also known as the relative complement or relative difference) denoted by $B \backslash A$ comprises the elements which are only present in B but not A (Figure 3.4). This can be expressed as:

$$B \backslash A = A^C \cap B \tag{3.7}$$

Where:

$$A^C = U \backslash A \tag{3.8}$$

defines the "complement" of set A: it comprises all the elements in the universal set that are not in A.

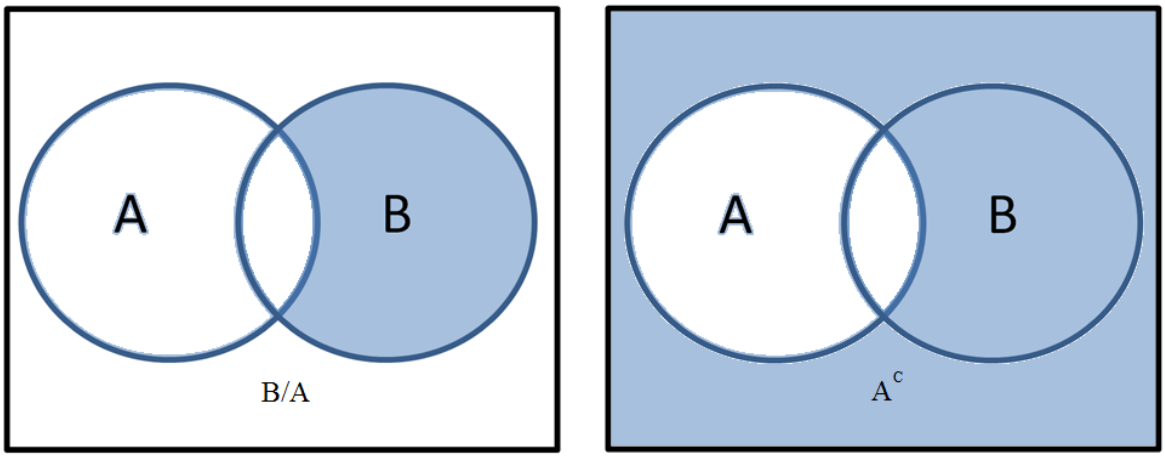

Fig. 3.4 Venn diagram representing set difference between sets

3.1.4 Symmetric Difference

The "symmetric difference" comprises all the elements which exist in all of the sets but not if they are shared in the intersections. It is denoted by the $\triangle$ operator (Figure 3.5):

$$A \triangle B$$
$$A \triangle B \triangle C \tag{3.9}$$

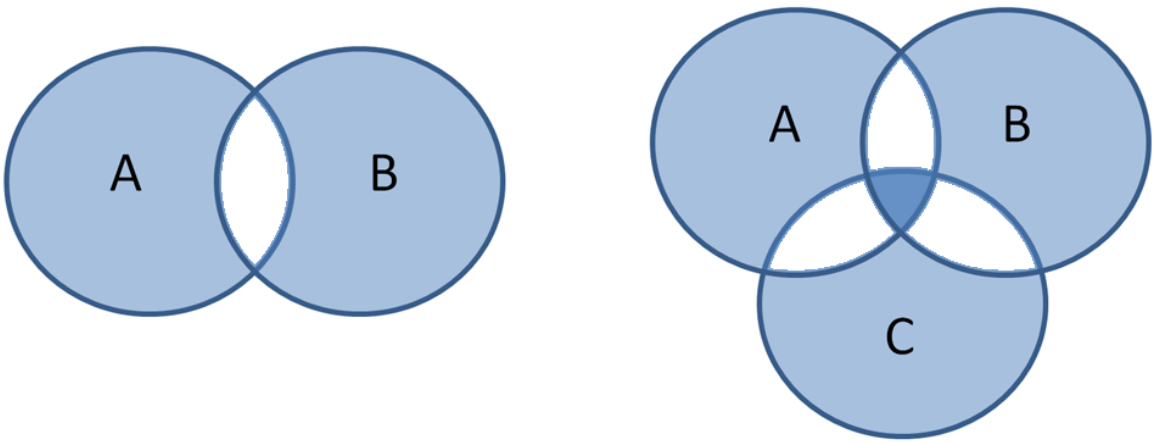

Fig. 3.5 Venn diagram representing the symmetric difference between sets

3.1.5 Cardinality of Sets

The cardinality of a set refers to the number of "elements in the set". It is sometimes called the “size” [74]. It refers only to the number of unique elements in the set. For example set B = {1, 2, 1, 6, 7, 8, 2} has a cardinality of 5. Cardinality is represented by vertical lines either side of the set definition/name [120].

$$\begin{aligned} A &= \{2,4,6\} & |A| &= 3 \\ A &= \{1,2,1,6,7,8,2\} & |B| &= 5 \end{aligned} \tag{3.10}$$

3.2 Binary Classification and ROC Curve Analysis:

3.2.1 Binary Classification

Binary (also known as binomial) classification is the dividing of data into two groups using a classification rule. Common binary classifications include medical testing to test and identify whether a patient has a particular disease, quality control to decide whether a new product is up to standard to be sold or spoilt, and information retrieval, in determining if information is relevant in a search result or not [140]. There could be serious repercussions if, in these examples, the classification were not accurate. Techniques used in binary classifiers include decision trees, Bayesian networks, support vector machines, neural networks, probit regression, and logit regression [140].

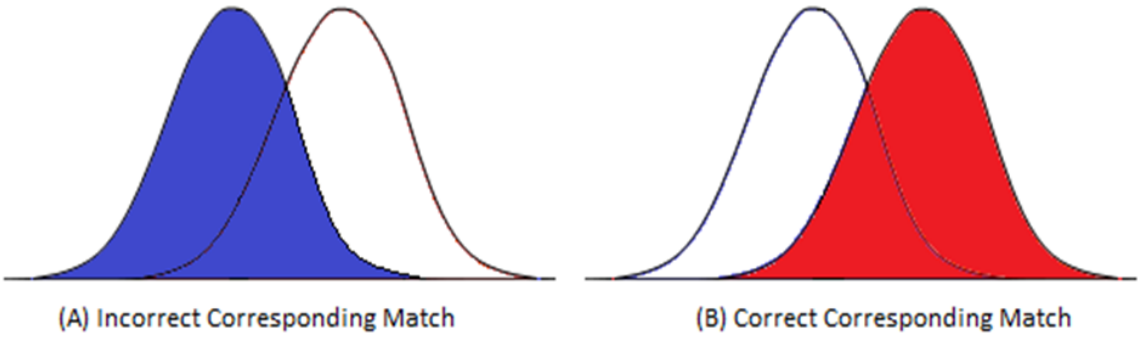

Fig. 3.6 Distribution of average binary outcomes where prediction and outcome agree (A) in blue, Distribution of average binary outcomes where prediction and outcome do agree (B) in red

In order to judge the performance of a binary classifier it is necessary to know to which of the two groups a data item belongs (Figure 3.6). A threshold or cutoff can be used for the prediction values and each data item can be assigned to one group or the other based on its prediction value and the cutoff (Figure 3.7).

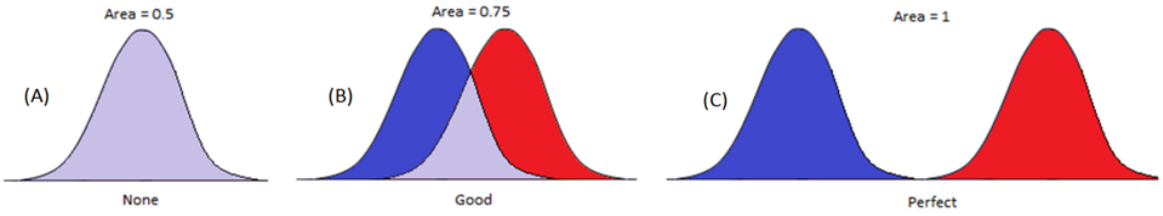

Fig. 3.7 (A) complete agreement between the results (B) good relationship between results (C) Perfectly separated results

Two sets of completely independent results will seldom have a textbook isolation between the two groups, i.e. there will be some overlap (Figure 3.8).

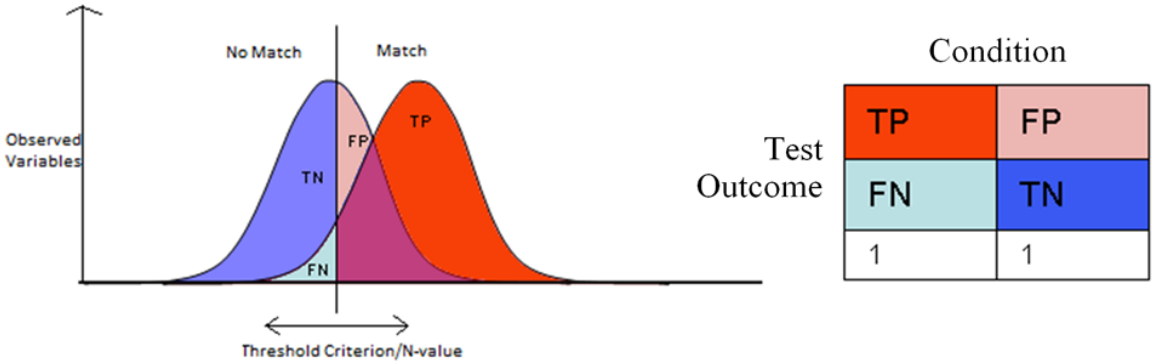

Fig. 3.8 Observed variable distributions vs. threshold criterion [132]

In order to explain the concepts involved we use the example of diagnosis of a disease using a medical test which can have two possible outcomes: 'positive' suggesting presence of the disease and 'negative' suggesting its absence. The test on an individual has a binary outcome: either positive or negative, but cannot have both. In order to evaluate the test it would need to be applied to a group of individuals known to have the disease and another group of individuals known to be free of the disease.

There are four possible outcomes for a binary classifier. A "True Positive" is sick individual correctly diagnosed as being sick. A False Positive is a healthy individual incorrectly diagnosed as sick. A True Negative is a healthy individual correctly diagnosed as healthy

and a False Negative is a sick individual incorrectly diagnosed as being healthy (Table **??**).

Test/Disease	Not Rejected	Rejected
With Disease (D = 0)	TN	FP
Without Disease (D = 1)	FN	TP

Table 3.1 Example of binary anlysis outcomes in Diseased vs. Observed Patient data

The number of True Positives/Negatives and False Positives/Negatives are taken at distinct intervals by varying the classifier prediction threshold. In a distribution diagram this amounts to moving it left or right. In the most extreme cases all patients are predicted to have the disease or conversely all patients will be predicted to be free of the disease (Figure 3.9).

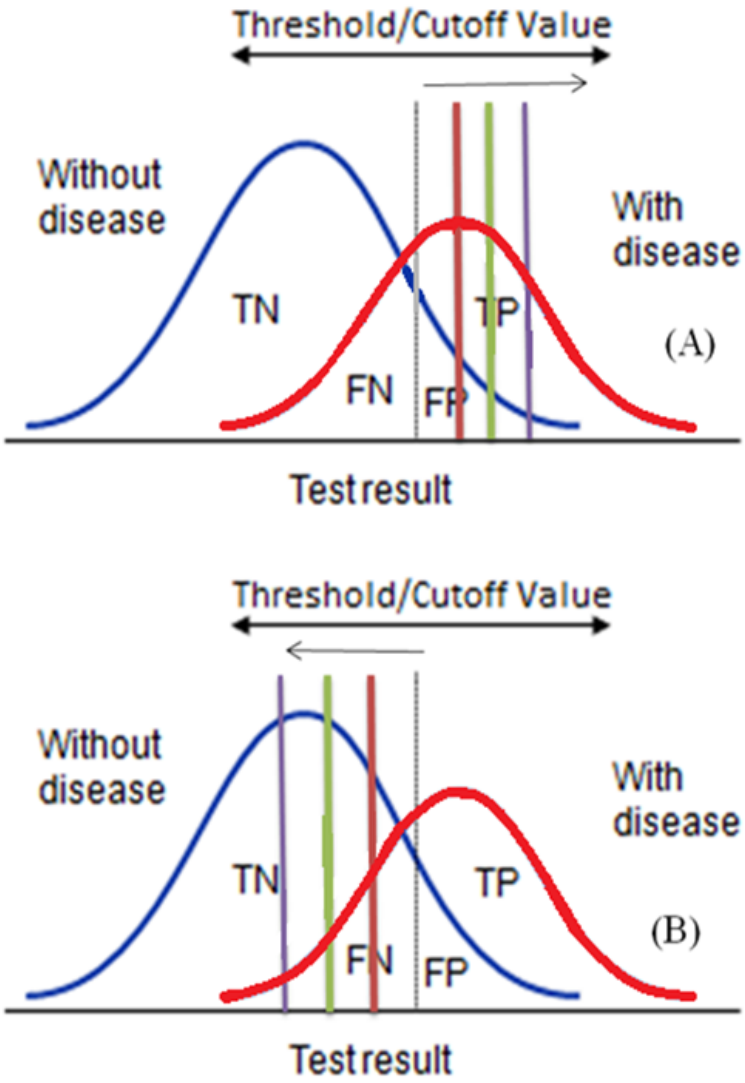

Fig. 3.9 Varying Threshold/Cut off value amongst the Test Data (A) Threshold value with increasing TP rate (B) Threshold value with increasing TN rate

The frequency for each of the four possible outcomes can be used to calculate the sensitivity and specificity:

$$Sensitivity = \frac{TP}{TP+FN} \qquad Specificity = \frac{TN}{FP+TN} \tag{3.11}$$

These statistical measures analyse the performance of the classification function. Sensitiv-

ity (also known as true positive rate) is the proportion of actual positives that are true positives (proportion of sick people correctly acknowledged as diseased) and is complementary to the false negative rate [140]. Specificity (also known as true negative rate) is the proportion of actual negatives that are true negatives (proportion of healthy people correctly identified as not diseased) and is complementary to the false positive rate (Figure 3.10).

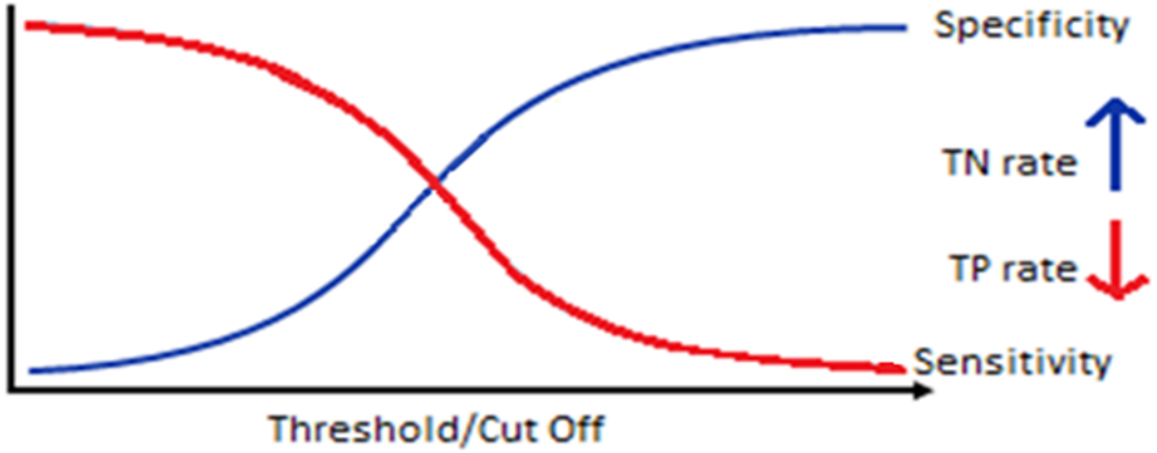

Fig. 3.10 Inverse proportionality between the True Negative and True Positive Rate

A faultless prediction result would show 100% sensitivity where all people from the sick group would be predicted as sick and 100% specificity where everyone from the healthy group would be predicted healthy. However any predictor will have some inherent error, which is known as the "Bayes error rate" [141]. There is often a trade-off between sensitivity and specificity [66]. Another useful measure is the "precision" or "positive predictive value" defined as the ratio of true positives to combined true and false positives. A high precision means we can have confidence that a positive prediction is an actual positive.

3.2.2 ROC Curve Analysis

An ROC (Receiver Operating Characteristic) curve is an essential tool for diagnostic sensitivity/specificity assessment, where the true positive rate (Sensitivity) is plotted against the false positive rate (1-Specificity) for different decision thresholds of a parameter. Each point on

the ROC curve signifies a sensitivity/specificity pair corresponding to a particular decision threshold [78].

The area under the ROC curve (AUC) is a measure of overall test performance and how well a parameter can distinguish between two diagnostic groups. An analysis with faultless discrimination (no intersection in the two datasets as seen in(Figure 3.7C)) displays an ROC curve, passing through the upper left quadrant (100% sensitivity, 100% specificity) demonstrating perfect discrimination between the two sets and would have an area of 1.0 [25]. The better the predictor the greater the AUC. A truly random predictor would have an AUC of 0.5 (Figure 3.11). In the disease example the AUC can be understood as the probability that the result from a randomly chosen diseased patient is more symptomatic of disease than from a randomly selected non-diseased individual [128]. This means it can be thought of as a nonparametric (no assumptions about the probability distributions of the variables being assessed) distance between disease/non-diseased test results. The problem with AUC is there is no clinically relevant meaning for this given example. It is only a theoretical statistic, which is also greatly affected by the range of large false positive values, which are often not particularly relevant [31].

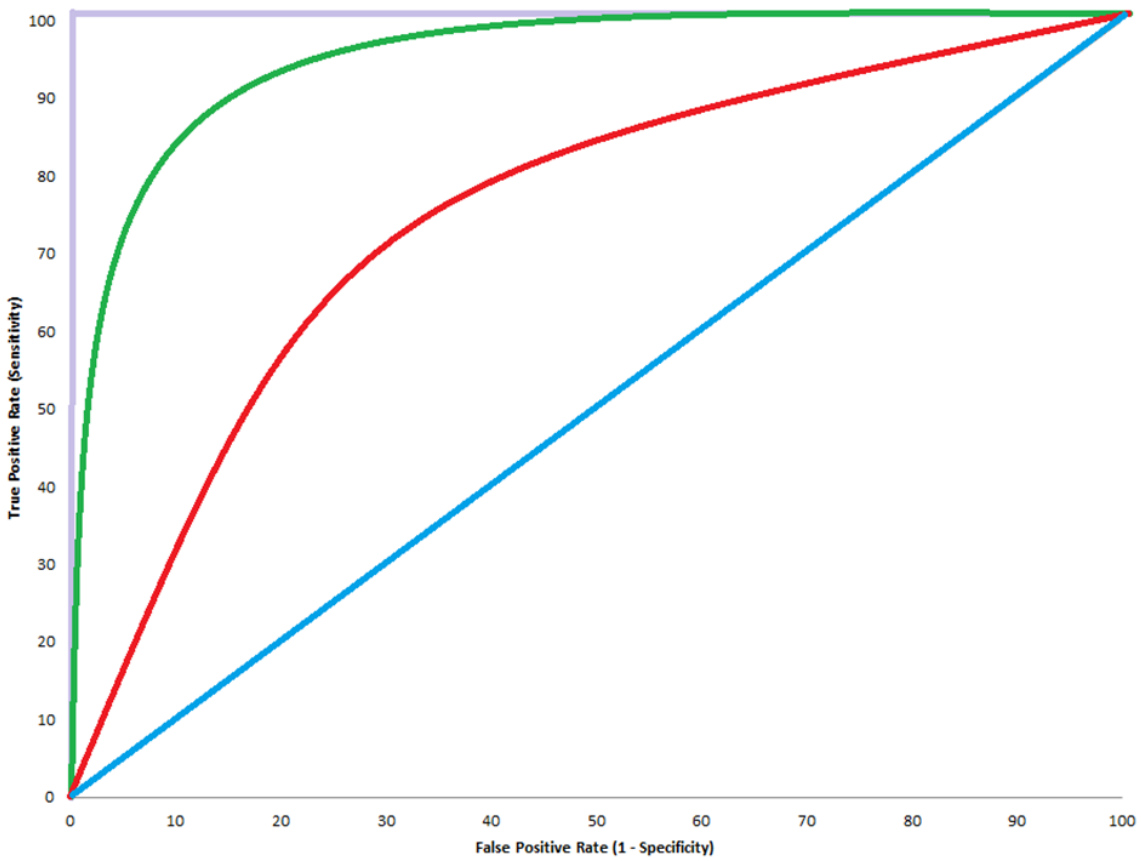

Fig. 3.11 Example of a strong correlation between both results ROC Curve, the purple line gives a perfect relationship, the green line indicates a good agreement, the red line shows a reasonable relationship and the blue line indicates a random guess.

3.3 Machine Learning:

The goal of Machine Learning can be characterized as the construction of computer systems that adapt and learn from experience. They are able to do this by discovering associations between input data and the classification for each data item encoded as an output value. If the relationship between these inputs and outputs is already known then machine learning is not required. However, this is rarely the case in real life systems [11]. Machine Learning can be broken down into two main approaches: supervised and unsupervised learning. In supervised learning training examples of known class (i.e. labeled training data) are used to construct a model for predicting a class for examples not present in the training data. In unsupervised learning the training data is unlabeled and the methods are used discover patterns that reflect the statistical structure of the overall collection of input patterns [83].

Linear regression uses least squares (estimating a quantity or fitting a function to data so as to minimize the sum of the squares of the differences between observed and estimated values) to find a best fitting hyperplane, producing coefficients which predict change in the dependent variable for one unit change in the independent variable. Logistic regression on the other hand, approximates the probability of an event occurring, so can be used to refer specifically to the problem in which the dependent variable is binary [90]. Predicting from knowledge of relevant independent variables does not give an exact numerical value of a dependent variable, but a probability (p) between 1 (incident happening) or 0 (incident not happening). In linear regression, the association between the dependent and the independent variables is linear; this assumption is not made in logistic regression because the “logistic regression function” is used instead, which can describe explanatory (predictor) variables, giving the probabilities of possible outcomes being modeled. A good example of this is in the American sport of baseball, where a “home run” is when a baseball player hits the baseball out of the arena the sport is being played in, scoring the best and maximum points available,

this is not an easy feat to accomplish. The "Hall of fame" highlights the best players of baseball since the game started, players have to be elected into it and it is considered to be the greatest honor bestowed on a professional player of the game. (Figure 3.12) highlights the correlation between the number of homeruns scored in a player's lifetime on the x axis and whether they have been inducted into the hall of fame (1 if they have 0 if not) on the y axis. Linear regression shows a probability of being elected to the hall of fame is less than 0 at approximately 250 home runs and an absolute certainty at approximately 625 home runs. Logistic regression offers a far more realistic and easily interpretable result [81].

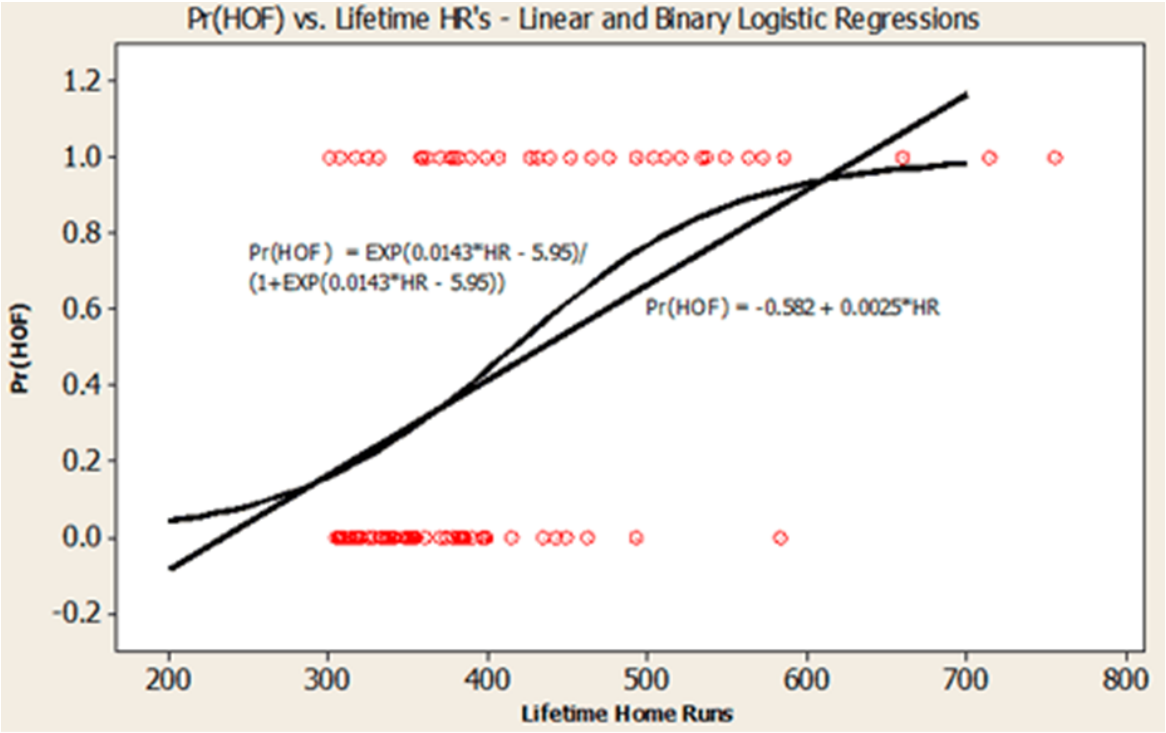

Fig. 3.12 Linear vs. Logistic Regression Analysis example: Hall of Fame (HOF) vs. lifetime home runs (HR) linear and binary logistic regressions [81]

Logistic Regression can also examine how well a model fits, and the significance of the relationships (between dependent and independent variables) being modeled. It is able to do this using probability scores as predicted values of the dependent variable [60]. It is used in estimating empirical values of the parameters in experimental data. An example of this

could be for analysing the relationship between a disease and age, where sampled individuals were examined for signs of disease being present = 1 and absent = 0. The mean and standard deviations of people who do, and do not show signs of disease would only give you half of the maximum age of each group, which would not be very accurate or scientific. If, however, individuals could be grouped into age classes and the percentage/proportion of these groups showing signs of disease could be examined, this would prove a great deal more accurate and useful. This would change the regression from a linear to a logistic regression analysis [37]. This is done by a logistic regression model:

$$P(Y|X) = \frac{e^{\beta_o+\beta_1 X}}{1+e^{\beta_o+\beta_1 X}}$$

This is the equivalent of saying:

Logit Transformation

$$ln\left(\frac{P(Y|X)}{1-P(Y|X)}\right) = \beta_o + \beta_1 X \tag{3.12}$$

The “logit transformation” represents the “logistic odds (odds ratio)” and is defined as the mathematical function of the inverse of the sigmoidal “logistic function/transform” when the function’s output is a probability p. It is also seen as a linear function of the independent variables. In the disease example, the odds of having the disease are calculated against there being a factor, present and absent:

	Risk Factor (X)	
Disease	Present (X = 1)	Absent (X = 0)
Yes (Y = 1)	P(Y = 1|X = 1)	P(Y = 1|X = 0)
No (Y = 0)	1 - P(Y = 1|X = 1)	1 - P(Y = 1|X = 0)

Table 3.2 Disease vs.. Risk Factor (X) odds table

The different outcomes can be tabulated to give the odds of disease being present and

absent which ultimately gives the odds ratio (OR).

$$\frac{P}{1-P} = e^{\beta_0+\beta_1 X} \tag{3.13}$$

Thus

$$\text{Odds Ratio (OR)} = \frac{\text{Odds for Disease with Risk Present}}{\text{Odds for Disease with Risk Absent}} = \frac{e^{\beta_0+\beta_1}}{e^{\beta_0}} = e^{\beta_1} \tag{3.14}$$

3.3.1 Regularization

One concern about machine learning techniques is that the training set data is too extreme for the test data to be classified with. It is also the case that a statistical model's can be affected by random error or noise, which does not give a true underlying statistical relationship. This generally occurs when a model is exceptionally complex, for example, when there are too many parameters, relative to the number of observations. This will produce a poor forecaster, by exaggerating errors or minor data variations. This is known as "overfitting" which occurs when a model starts to memorize training data rather than learning general trends. If the number of parameters is equal or greater than the number of observations, a simple learning process model can perfectly forecast the training data by remembering all of the training data, but it would not be able to predict new or yet unseen data, because the model has not been taught to generalize (Figure 3.13).

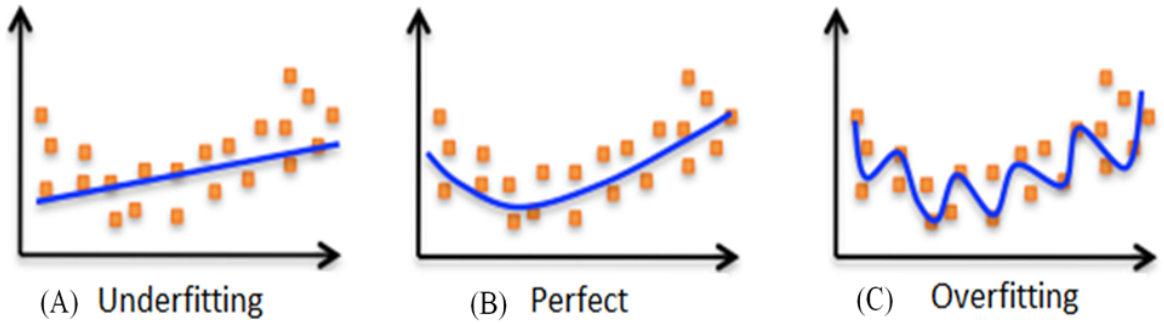

Fig. 3.13 Examples of data fitting (A) A linear function or too few features set gives a model which under fits the data (B) ideal fitting of model (C) A polynomial function or large set of features when fitted into a model will over fit on the data [42]

3.3.2 Bayesian Probability

To avoid overfitting, supplementary methods such as cross-validation, regularization, Bayesian priors on parameters or model comparison provide supplementary information that can show further training does not produce a better generalization [125]. This can be achieved by the above methods in one of two ways, firstly by penalizing excessively complex models and secondly by testing the model's generalization capability by calculating its performance on a data set not used in training, which will approximate the archetypal unseen data that a model will analyse [11]. To correct overfitting a constraint can be introduced on the overall magnitude of the parameters. This can be done by "Regularization", so called because a regularizer attempts to keep parameters more normal/regular [127]. It is a bias on the model, which forces the learning to favor certain types of data points over others because of its penalized score system, otherwise known as a "loss function" (charts an event/value of variables with a number, signifying a "cost").

To improve the predicting capability of a machine learning program additional screening measures can be employed to maximize the previous knowledge from the training data. This can be done by using Bayesian learning which allows a combination of observed data

and prior knowledge to provide practical learning algorithms. It is a generative (model based) approach, which gives a conceptual framework. The significance of this is that any kind of object (time series, trees, etc.) can be classified, founded on a probabilistic model specification [111]. A Bayesian probability can be seen as the degree of believability of a proposition, given the requirement that probabilities are prior beliefs conditioned on data and this “is optimal”, given a good model, a good prior and a good loss function. The term posterior probability of a random/uncertain event is the conditional probability that is allocated after the relevant evidence is acquired and taken into account and considered [7]. The posterior probability distribution is the probability distribution, provisional on the evidence/data then seen and acquired. "Posterior", in this framework, refers to considering the relevant evidence related to the specific case being inspected [38]. The data/evidence is given by:

$$p(Y) \tag{3.15}$$

The prior assumption/knowledge is given by:

$$p(\theta) \tag{3.16}$$

The likelihood is seen as the intersection of the evidence with the prior assumption and is given by:

$$p(Y|\theta) \tag{3.17}$$

And the posterior or outcome is given by:

$$p(\theta|Y) \tag{3.18}$$

(Figure 3.14) graphically depicts the relationship between prior knowledge and its ability

to predict an outcome in data.

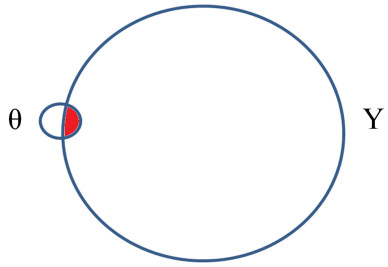

Fig. 3.14 Venn diagram highlighting the intersect between the data and prior which gives the likelihood.

It is generally the case that the most probable hypothesis is favored given the training data. This is a useful observation because it does not depend on $p(Y)$. Baye's rule is given as:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)} \tag{3.19}$$

This can also be written as:

$$p(\theta|Y) \propto p(Y|\theta)p(\theta) \tag{3.20}$$

Bayes theorem allows prior knowledge to be incorporated into computing statistical probabilities, as regularization can have little effect, this can instead be interpreted as a Bayesian prior over the weight, which gives a preference for weights of small magnitude giving improved performance.

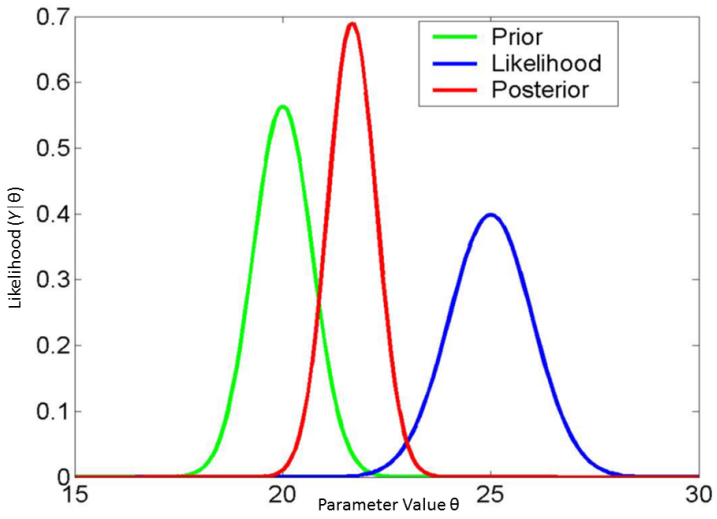

Fig. 3.15 Venn diagram highlighting the intersect between the data and prior which gives the likelihood.

The Gaussian distributions of the elements from Bayes rule (Figure 3.15) highlight that the posterior probability of the parameters given the data is an optimal combination of prior knowledge and new data weighted by their relative precision.

3.3.3 Logistic Regression

Given labelled training data $D = \{(\mathbf{x}_i, t_i)\}$, regularised logistic regression constructs a decision rule that can be used to distinguished between objects belonging to two classes. $\mathbf{x}_\mathrm{i}$, represents a vector of attributes describing the i^{th} example and t_i indicates the class to which it belongs ($t_i = 1$ for the positive class and $t_i = 0$ for the negative class) The logistic regression model is of the form:

$$logit(y(x)) = \mathbf{w} \cdot \mathbf{x} + b \text{ where } logit(p) = \log\left(\frac{p}{1-p}\right) \tag{3.21}$$

and $\mathbf{w}$ is a vector of regression coefficients. The optimal value of the regression coeffi-

cients is determined by minimising the regularised cross-entropy training criterion:

$$E = \frac{1}{2}\|\mathbf{w}\|^2 - \frac{\gamma}{2}\sum_{i=1}^{L}\left[t^i \log\left(y^i\right) + \left(1-t^i\right)\log\left(1-y^i\right)\right] \tag{3.22}$$

Where $y_i = y(\mathbf{x}_i)$, L is the total number of items in the dataset and γ is a regularization parameter governing the bias-variance trade-off. The output of the logistic regression model can then be regarded as an estimate of the Bayesian a-posteri probability of class membership, i.e.

$$y(x) \approx P(t = 1|x) \tag{3.23}$$

The optimal value for the regularization parameter γ was efficiently determined by minimising the leave-one-out cross-validation estimate of the test cross-entropy.

3.4 Graph Theory:

Graph theory is the mathematical study of points and lines [16, 17] . In particular, it involves the ways in which sets of points, called vertices, can be connected by lines or arcs, called edges (Figure 3.16) [72]. Graphs in this context differ from the more familiar coordinate plots that portray data correlations [18, 122].

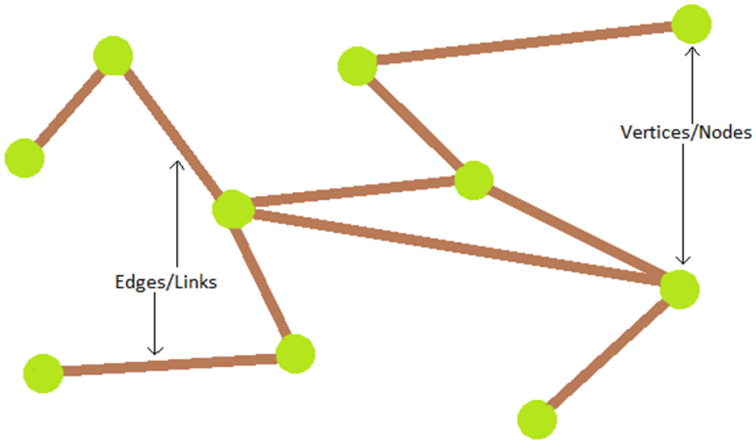

Fig. 3.16 Diagram highlighting Graph Theory glossary

Graphs are classified according to their complexity, the number of edges allowed between any two vertices, and whether or not directions are assigned to edges (Figure 3.17). Various sets of rules result in specific properties that can be stated as theorems [44].

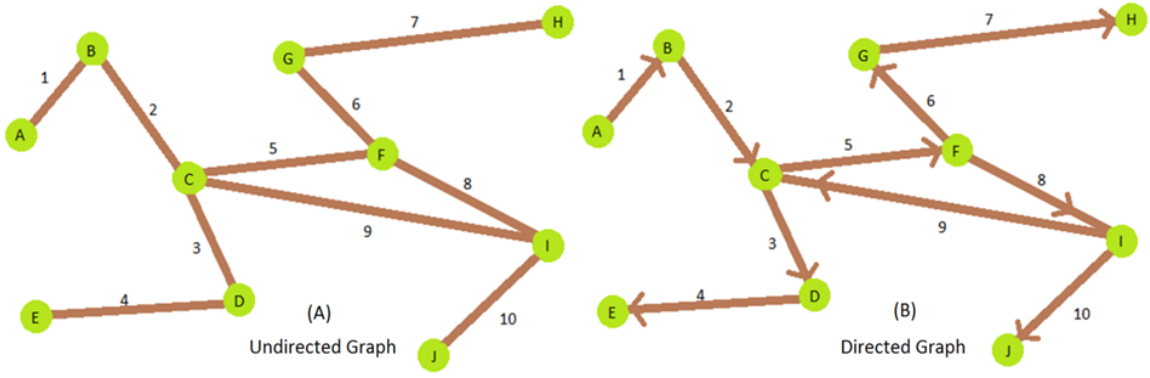

Fig. 3.17 (A) Undirected graph (B) Directed graph

A graph can be detailed mathematically as $G = (V, E)$; consisting of a set of vertices V

and a set of edges E. The edges are 2-element subsets of V. The vertices belonging to an edge are called the ends, endpoints, or end vertices of the edge. They can exist without belonging to an edge. Formally an edge is denoted as $\{u,v\}$ where u and v denote vertices, but this can be abbreviated to uv. The order of a graph is defined mathematically as $|V|$ which gives the number of vertices present. The graph's size is given as $|E|$ which provides the number of edges [130].

Given $G = (V,E)$ is a graph, a path of length k from a vertex u to a vertex v is a sequence $u = v_0, v_1, v_2, ..., v_k = v$ of vertices such that $(v_{i-1}, v_i) \in E$, and there are no repeated edges. Repeated vertices are allowed. If there are parallel edges in the graph, then the sequence must specify, for each i, which of the edges (v_{i-1}, v_i) is in the path. A simple path is a path with no repeated vertices. In this example the path goes from vertex A to J [134].

$$\{v_A, v_B, v_C, v_I, v_J\} \tag{3.24}$$

The degree of a vertex is the number of edges that are adjacent to it; for example vertex F in (Figure 3.17) has a degree of 3, because of the 3 edges which connect it. A graph is called connected if for every two vertices u and v there is a path from u to v. Otherwise the graph is called disconnected [130]. A cycle (also known as a circuit) is a path from $v \in V$ to itself with at least 3 vertices where the first and the last vertex in the path are adjacent with at least one edge. The single-edge case is called a loop. A simple cycle is a cycle with no repeated vertices except for the first and the last vertex. For example in (Figure 3.17) vertices C, F and I are in a cycle. A tree is a graph that is connected and does not contain a cycle. For example when the $\{v_C, v_I\}$ edge is removed the graph becomes a tree (Figure 3.18). The term "root" node refers to the beginning node (starting point) of the tree where the path can be plotted from [47].

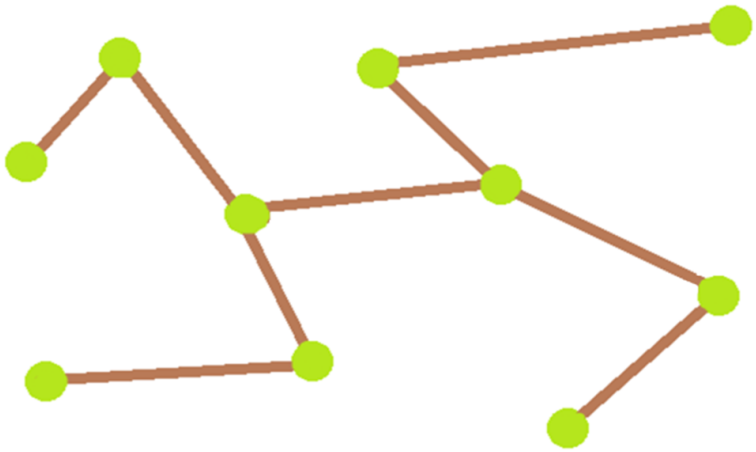

Fig. 3.18 An example of a tree would be a phylogeny tree, used primarily in work done in evolutionary relationships.

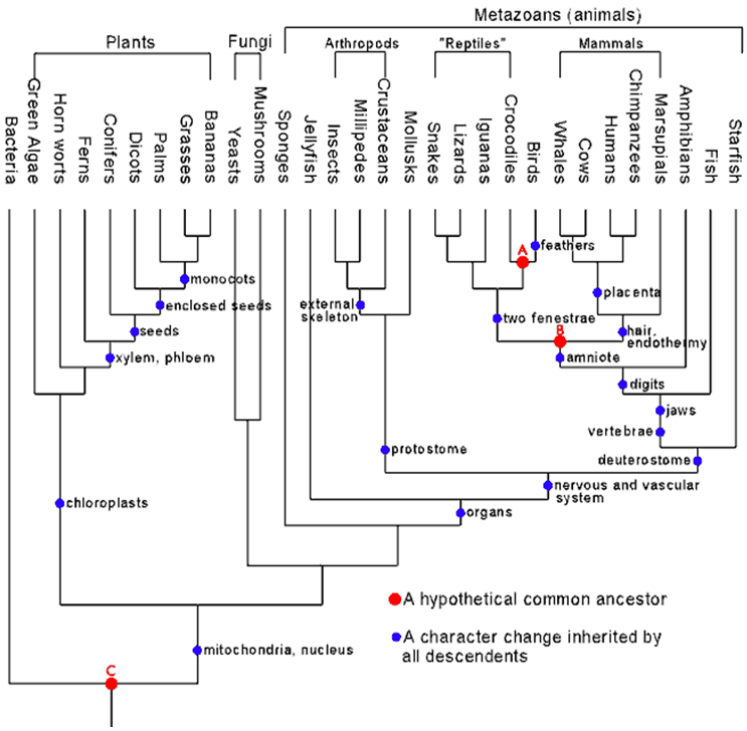

Fig. 3.19 Evolutionary tree of life with graph theory [126]

3.4.1 Graph Traversal

Graphs are used to represent many types of relationships and procedures allowing theories and practical problems to be characterised visually. In mathematics, graphs can be used in

geometry and topology [100]. It is also strongly associated with group theory. In computing, graphs are utilized to symbolize the flow of information, networks of communication, and data organization (Figure 3.19). The development of algorithms to analyse graphs is an important area of computer science. It is often the case that a graph must be simplified to decipher its meaning [134]. Many mathematical problems require the “traversal” of a graph, which means to travel from one vertex to another by way of the edges.

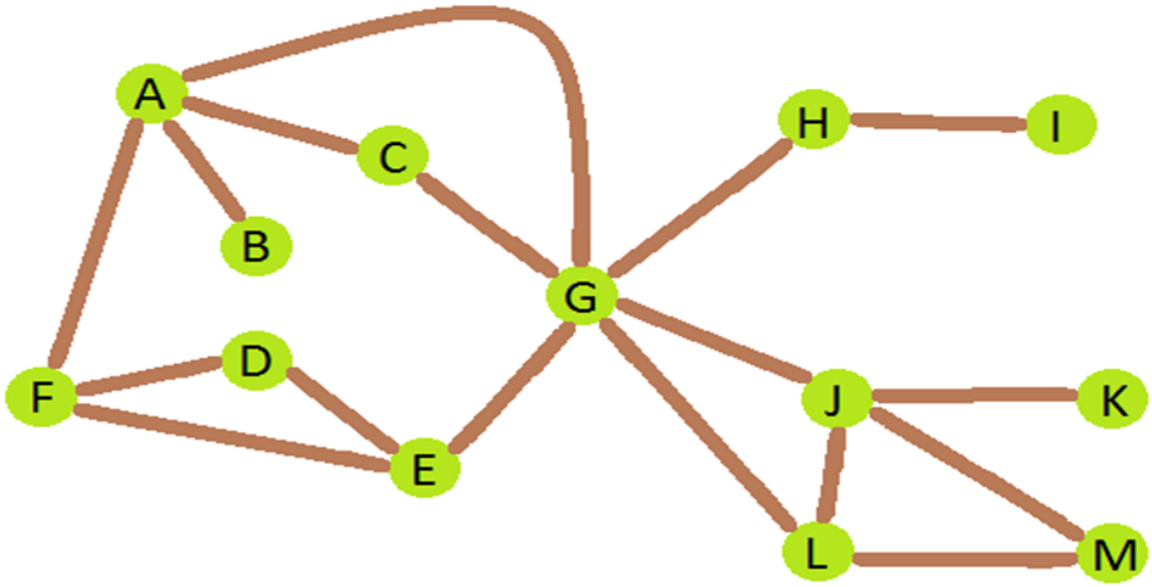

Fig. 3.20 An example graph

Graph traversal is a discipline in graph theory which looks at visiting all nodes in a graph in a specific way, updating and/or examining weights (if it is a weighted graph). Tree traversal is a distinct class of graph traversal [130]. The journey to each node of the graph might involve more than one visit because it is not a given that the node has been visited already. As graphs become denser, this redundancy becomes more widespread, causing CPU time to escalate dramatically. It is important to remember which vertices have previously been visited by the algorithm, so that vertices are returned to infrequently (or an infinite loop is not created where the traversal would continue forever). This is possible by initially marking each vertex with a "colour" or "visitation" which can then be checked and rationalised as each

vertex is visited [5]. If already visited, then it is disregarded and the path stops; otherwise the algorithm checks/updates and remains on its present path. In some circumstances the visiting and recording of the vertex on the graphs is not required [117]. This is the case with a tree and the Depth-First Search (DFS) and the Breadth-First Search (BFS) algorithms. (Figure 3.20) is an example graph, used for demonstrating the graph traversal methods [134].

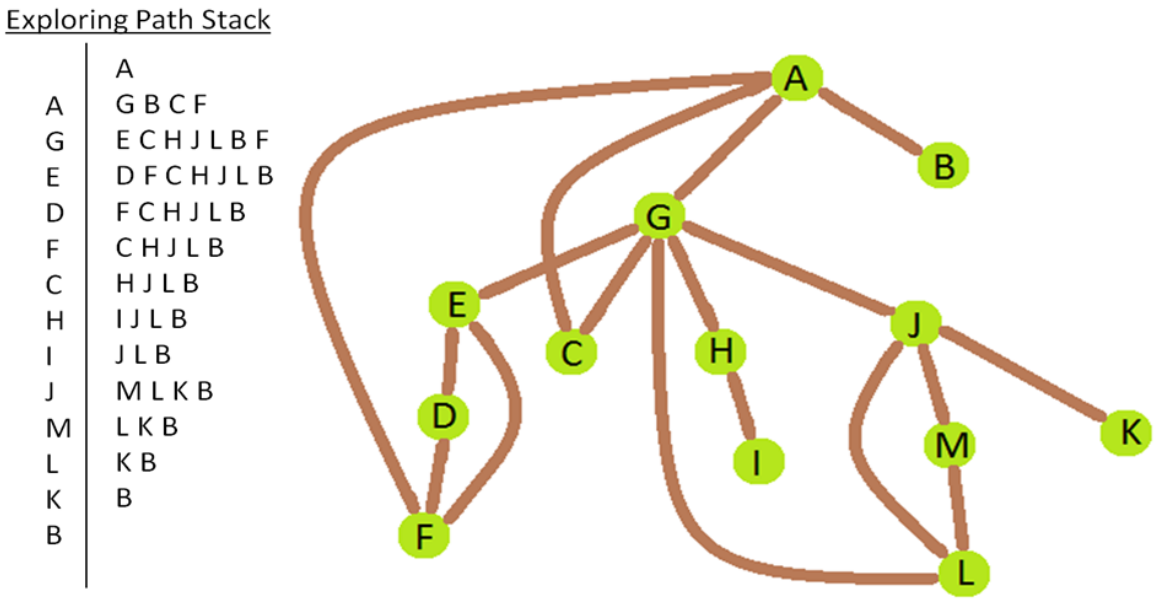

Fig. 3.21 Example Graph explored by DFS

DFS is an algorithm which produces a path that progresses through the tree by starting at the root vertex, following each edge to the next vertex systemically. It proceeds by going deeper into the tree until an objective vertex is found, or an end vertex with no further edges to other vertices is visited (Figure 3.21). The search then reverses, returning to the most recent vertex with an edge it has yet to travel [129]. A counting stack is used to determine the vertices the search meets along the way. The amount of time a search algorithm takes to transverse a graph is of huge significance to the application and nature of the problem. DFS is typically used to traverse a whole graph, and takes time $O(|V|+|E|)$ with the number of vertices $(|V|)$ and number of edges $(|E|)$ so is linear in graph size. The memory usage is $O(|V|)$ as a worst case scenario to store the stack of vertices on the current search path and the

already-visited vertices set. For real-world problems the size and nature of the graph might make DFS algorithm unattractive due to memory and time limitations [28]. To get around this, the search can be limited to a certain depth. DFS is able to accommodate heuristic techniques such as the "iterative deepening depth-first search" method whereby the depth of the search is expanded on each iteration. When a suitable depth limit is not known, a "priori" (some knowledge of the problem is already known) iterative deepening depth-first search uses DFS iteratively with an expanding factor greater than one. This means iterative deepening escalates running time by only a constant factor rather than where the correct depth limit is anticipated, where there is a geometric increase of the number of vertices per depth level [44].

Breadth-first search (BFS) is an exhaustive uninformed search algorithm; it is also non-heuristic, beginning at the root node (Figure 3.22). It methodically investigates each of the nearest vertices. It searches the entire graph until it identifies the vertex it is seeking [80] or visits all the vertices. All child vertices (vertices further along the path) explored are added to a "First In, First Out queue" (where the oldest entry, or bottom of the stack, is handled first, namely first-come, first-serve behavior). Vertices that have not yet been visited for their neighbours are stored/recorded (in a queue or linked list). Once visited they are logged and thus recorded [69].

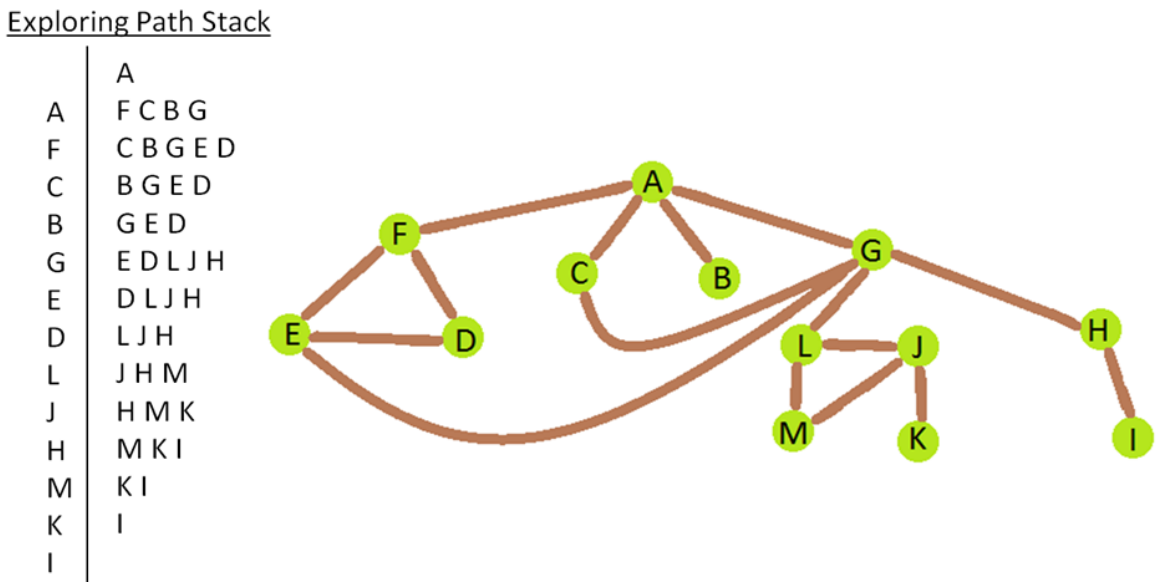

Fig. 3.22 Example Graph explored by BFS

The decision as to whether to use depth or breadth first searches largely depends on the different properties of vertex ordering produced by the two algorithms, rather than their complexity, as space and time constraints are similar for both (Figure 3.23).

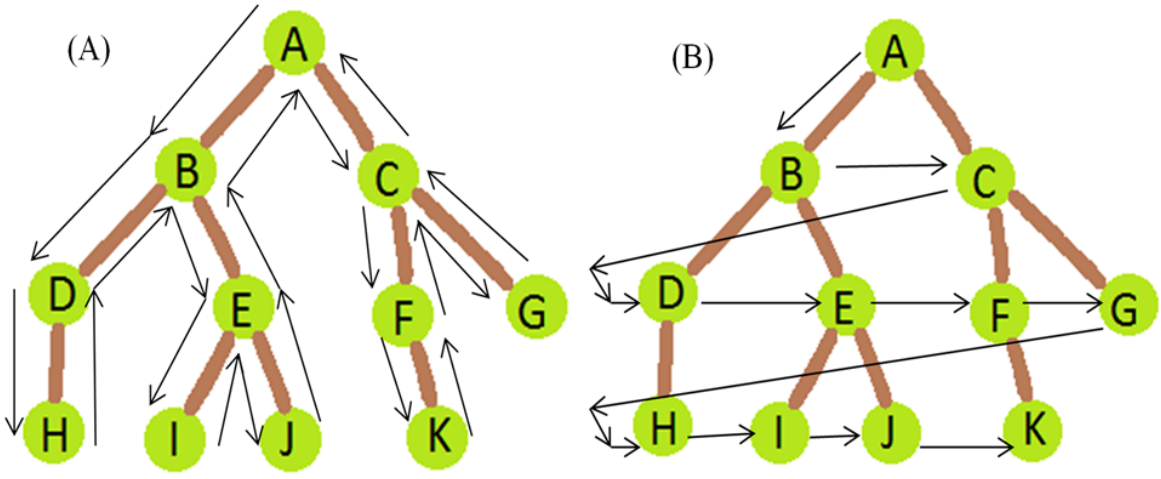

Fig. 3.23 Comparison between (A) DFS and (B) BFS

3.4.2 Graph Theory Algorithms & Problem Solving Applications

The inter-connection within the graph is of huge importance in the field of Graph Theory as it can often have a direct effect on the paths taken around a graph. Overarching graphs can be made from several smaller graphs (subgraphs). Tarjan's Algorithm is an algorithm which looks for "strongly connected components of a graph". It is akin in efficiency to another algorithm called the "path-based strong component algorithm" which looks for strongly connected components of a directed graph by using a depth-first search, amalgamated with two stacks, one monitoring the vertices in the current component and the second remembering the current search path [123]. Tarjan's algorithm uses directed graphs; any vertex that is not on a directed cycle forms a strongly connected component on its own. It divides the graph's vertices into the graph's strongly connected components, with a minimum of one vertex of the graph appearing in one of the strongly connected components [123].

First, a depth-first search begins from a random vertex (successive depth-first searches start on any vertices not yet discovered). The DFS traverses every vertex once, not returning to any vertex already found. This gives a spanning forest of search graphs, which in turn highlight the strongly connected components, emphasised by particular subtrees in the spanning forest. The start vertices of these subtrees are called the "roots" of the strongly connected components. The random nature of this algorithm means any vertex of a strongly connected component could potentially function as the root. The end result is a picture of the connectedness of the graph and the number of subgraphs present [5]. This algorithm is an inbuilt function, "conncomp" of Matlab, a high-level language and interactive environment for numerical computation, visualization, and programming (Figure 3.24).

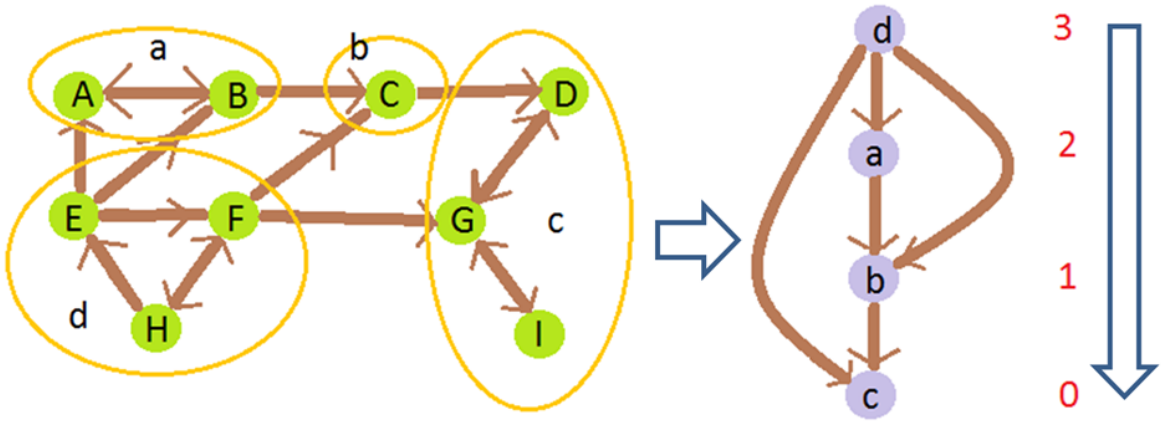

Fig. 3.24 Example of the Conncomp algorithm, indicating the degree of connectedness, the higher the number the more connected the subgraph.

In addition to looking at the routes taken through a graph and the overall breakdown and way a graph can be constructed, Graph theory can also be utilized for decision making, high level processing and basic artificial intelligence in the form of "decision trees" [21]. Branch and bound algorithms are an example of adaptive partition strategies to solve global optimization models (solving a problem in the best possible way). Branch refers to deriving new sub problems which lead to the so-called branch decision tree in the partition [68]. The bound denotes the decision made and how to proceed to the next vertex, this is done by calculating the upper and lower bounds for the minimum value, where a "lower" bound is said to be more preferable, by reducing the search space and finding the optimal result.

These algorithms use an exhaustive search method (every vertex is visited and checked before a decision is made) with partition, sampling, and successive lower and upper bounding procedures iteratively on active and deeper subsets (Figure 3.25). Branch and bound incorporates many approaches, allowing for various implementations, depending on some a-priori structural knowledge [44, 75]. This allows general branch and bound methodology to be applicable to a broad class of global optimization problems through a process of decision tree choices [117].

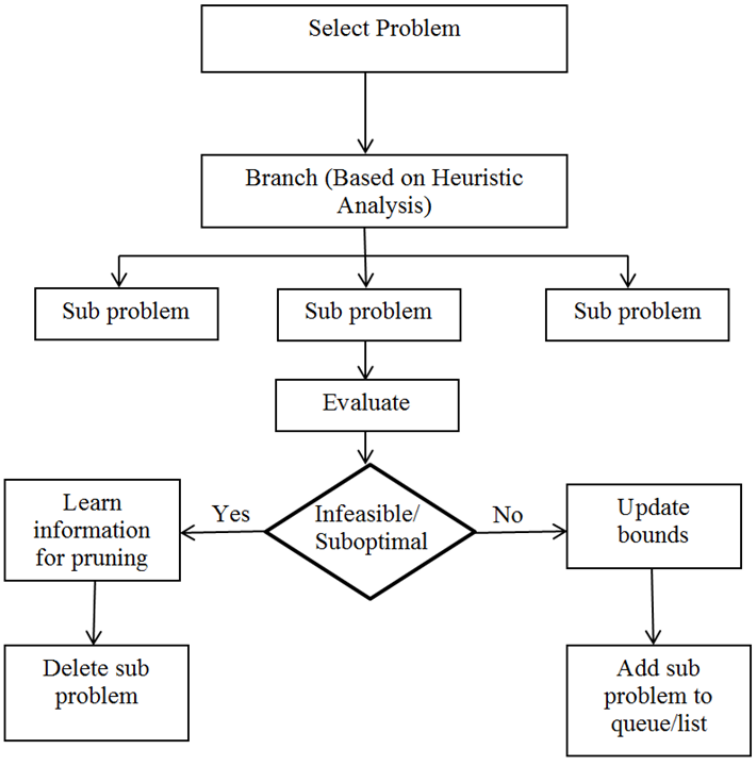

Fig. 3.25 Branch and Bound Algorithm flow diagram/decision tree

Chapter 4

Results: Residue Based Contact Analysis

4.1 Atom Based Contact Analysis

Our analysis uses the idea of the interdomain amino acid residue contacts. A "contact" between residue i and j means any heavy atom of residue i is within 4Å of any heaving atom of residue j. Excluded from this analysis are residues in the bending regions and those residues within 5.5 Å of the hinge axis as described in Methodology (Figure 4.1). The amino acids making contact can be identified and categorized by whether they are preserved or unique to each conformation.

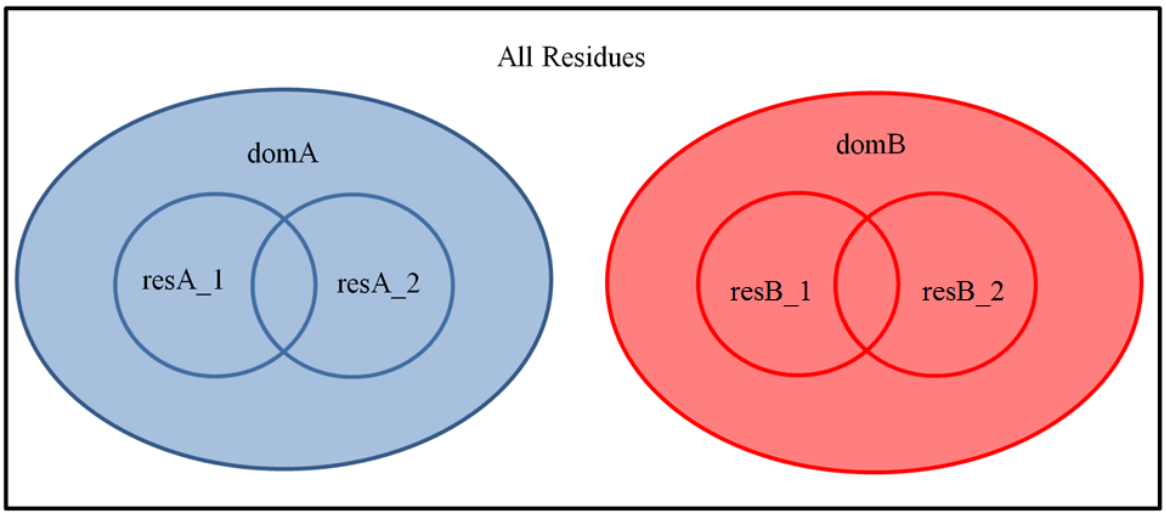

Fig. 4.1 Venn diagram representation of all residues in both domains and those that contact the other domain

Using these residue numbers as identifiers, set theory can be applied to specify whether particular contacts are preserved or unique between conformations. We define the following sets:

$\{\text{resA\_1}\}$ = the set of residue numbers for residues in domain A that contact residues in domain B in conformation 1.

$\{\text{resB\_1}\}$ = the set of residue numbers for residues in domain B that contact residues in domain A in conformation 1.

$\{\text{resA\_2}\}$ = the set of residue numbers for residues in domain A that contact residues in domain B in conformation 2.

$\{\text{resB\_2}\}$ = the set of residue numbers for residues in domain B that contact residues in domain A in conformation 2.

(4.1)

These residue sets once stripped of the excluded residues can be represented by the

domain they are assigned by DynDom (Figure 4.2).

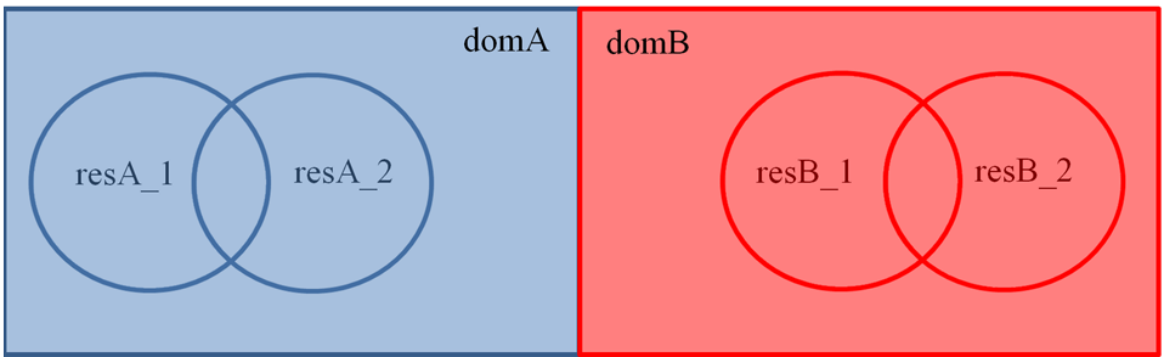

Fig. 4.2 Venn diagram for all residues in Domains A and B

The preserved sets for each domain, denoted as {resA_preserved} and {resB_preserved} are residues which maintain contacts between the two conformations and are given as:

$$\{\text{resA\_preserved}\} = \{\text{resA\_1}\} \cap \{\text{resA\_2}\} \tag{4.2}$$

$$\{\text{resB\_preserved}\} = \{\text{resB\_1}\} \cap \{\text{resB\_2}\} \tag{4.3}$$

This can be illustrated by the intersection of the two conformations from each domain (Figure 4.3).

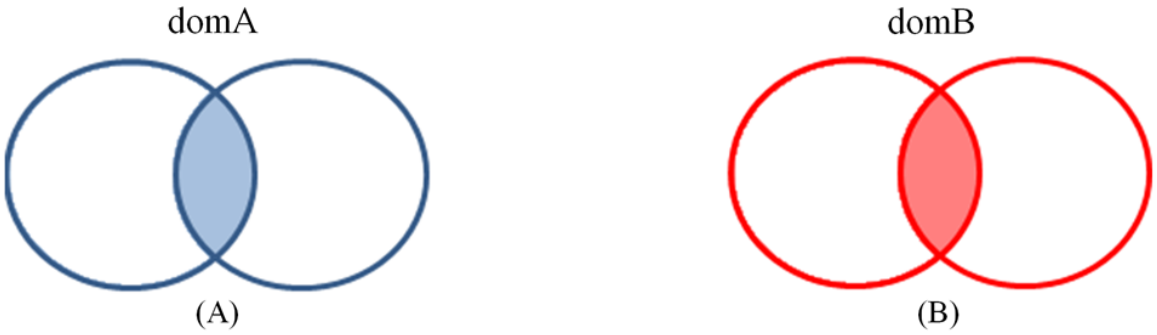

Fig. 4.3 Intersection of Preserved Residues of (A) Domain A coloured blue and (B) Domain B coloured red highlight the preserved residues between the conformations.

The preserved set for both Domain A and B denoted {res_preserved} is given as:

$$\{\text{res\_preserved}\} = \{\text{resA\_preserved}\} \cup \{\text{resB\_preserved}\} \tag{4.4}$$

The set of unique contacts in domain A conformation1 and domain B conformation 1, denoted as {resA_unique_1} and {resB_unique_1} are defined by the set difference and given by for Domain A and B in conformation 1 respectively as

$$\{\text{resA\_unique\_1}\} = \{\text{resA\_1}\} \backslash \{\text{resA\_2}\} \tag{4.5}$$

$$\{\text{resB\_unique\_1}\} = \{\text{resB\_1}\} \backslash \{\text{resB\_2}\} \tag{4.6}$$

Which can be depicted as a Venn diagram according to the domains they represent (Figure 4.4):

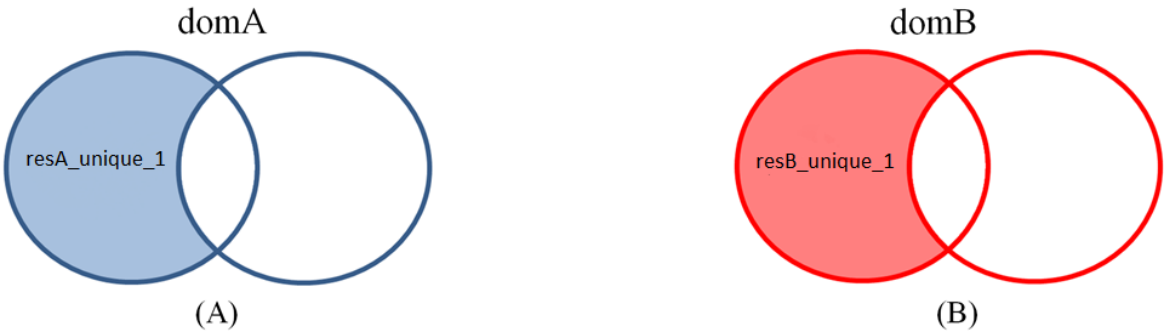

Fig. 4.4 Set Difference between the first conformation of each domain highlights the unique residues in (A) Domain A shaded in blue and (B) Domain B shaded in red.

The separate unique sets for each domain can then be combined to give the set of unique contacts in one conformation, but not the other. In the case of conformation 1 this is defined as {res_unique_1} given as:

$$\{\text{res\_unique\_1}\} = \{\text{resA\_unique\_1}\} \cup \{\text{resB\_unique\_1}\} \tag{4.7}$$

The same can be applied to conformation 2 {resA_unique_2} and {resB_unique_2} the set of residues in domain A and domain B, respectively, making contact in conformation 2 but not in conformation 1 are given as:

$$\{\text{resA\_unique\_2}\} = \{\text{resA\_2}\} \backslash \{\text{resA\_1}\} \tag{4.8}$$

$$\{\text{resB\_unique\_2}\} = \{\text{resB\_2}\} \backslash \{\text{resB\_1}\} \tag{4.9}$$

This can be depicted in a Venn diagram in (Figure 4.5).

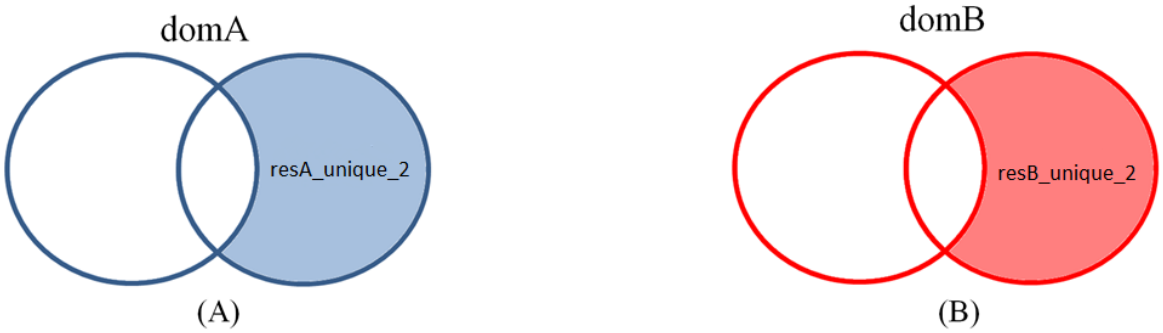

Fig. 4.5 Set Difference between the second conformation of each domain highlights the unique residues in (A) Domain A shaded in blue and (B) Domain B shaded in red.

The total number of unique contacts in conformation 2 denoted {res_unique_2} is given by:

$$\{\text{res\_unique\_2}\} = \{\text{resA\_unique\_2}\} \cup \{\text{resB\_unique\_2}\} \tag{4.10}$$

An alternative approach is to use the symmetric difference to get the number of unique contacts for each domain in both conformations defined with symmetric difference function:

$$\{\text{resA\_unique}\} = \{\text{resA\_1}\} \Delta \{\text{resA\_2}\} \tag{4.11}$$

$$\{\text{resB\_unique}\} = \{\text{resB\_1}\}\Delta\{\text{resB\_2}\} \tag{4.12}$$

The symmetric difference between these unique sets can also be shown as a Venn diagram:

Fig. 4.6 Symmetric Difference between both conformations of each domain highlights the unique residues in (A) Domain A shaded in blue and (B) Domain B shaded in red.

The set of unique contacts {res_unique} can then be taken as:

$$\{\text{res\_unique}\} = \{\text{res\_unique\_1}\} \cup \{\text{res\_unique\_2}\} \tag{4.13}$$

or

$$\{\text{res\_unique}\} = \{\text{resA\_unique}\} \cup \{\text{resB\_unique}\} \tag{4.14}$$

where

$$\{\} = \{\text{res\_unique\_1}\} \cap \{\text{res\_unique\_2}\} \tag{4.15}$$

$$\{\} = \{\text{resA\_unique}\} \cap \{\text{resB\_unique}\} \tag{4.16}$$

These sets defined can be used for counting the number of contacts between the domains. Then number of residues that maintain contact between the domains across the conformational

change, defined as N_p:

$$Np = |\{\text{res\_preserved}\}| \tag{4.17}$$

Number of new interatomic contacts between the domains in conformation 1:

$$N_{u1} = |\{\text{res\_unique\_1}\}| \tag{4.18}$$

Number of new interatomic contacts between the domains in conformation 2:

$$N_{u2} = |\{\text{res\_unique\_2}\}| \tag{4.19}$$

Number of new interatomic contacts between the conformations in domain A:

$$N_{uA} = |\{\text{resA\_unique}\}| \tag{4.20}$$

Number of new interatomic contacts between the conformations in domain B:

$$N_{uB} = |\{\text{resB\_unique}\}| \tag{4.21}$$

Number of total new interatomic contacts between the conformations in both domains:

$$N_u = |\{\text{res\_unique}\}| \tag{4.22}$$

or equivalently:

$$N_u = N_{u1} + N_{u2} = N_{uA} + N_{uB} \tag{4.23}$$

4.2 N-Value Calculation plotting against Angle of Rotation

The numbers of preserved and unique contacts (as defined in the previous section) allow us to define a measure, the "N value" given as:

$$N_{value} = \frac{N_p}{N_p + N_u} \tag{4.24}$$

The N value as the nice property of being equal to 1 when all contacts are preserved and there are no unique contacts and being equal to 0 when all contacts are unique and none are preserved. N value was calculated for all domain movements in the NRDB2d database. When N value is plotted against the angle of rotation (calculated by DynDom) for each protein in the NRDB2d (1822 in total), one can discern a general trend for the rotation angle to increase with decreasing N value (Figure 4.7).

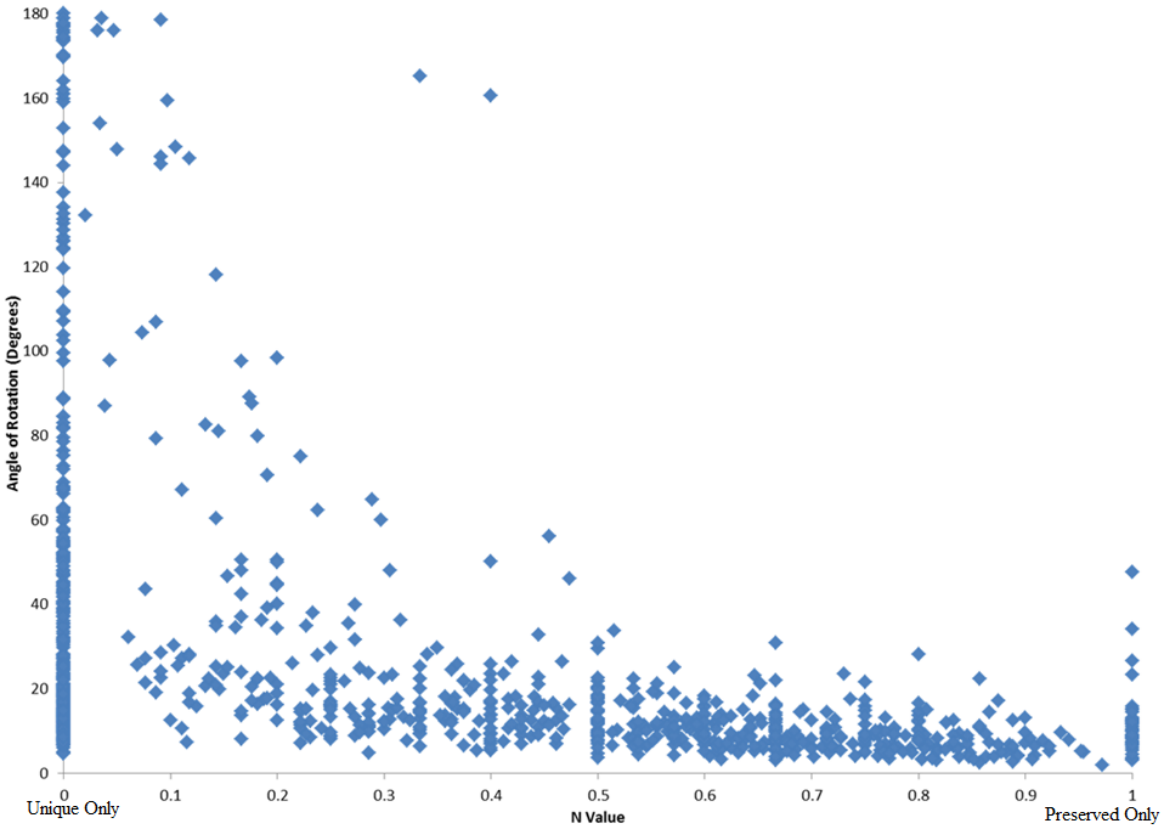

Fig. 4.7 Angle of Rotation vs. N-value XY Scatter Graph.

According to the classification scheme of Gerstein et al. [40] which assigns protein domain movements into two main categories hinge and shear one would expect a N value close to 1 (where there would be no new contacts made) for a shear motion and a N value close to 0 (with no preserved contacts) for a hinge movement.

4.3 Correspondence between DBMM and NRDB2d

Gerstein et al's method for classifying domain movements in protein is presented on the DBMM website (as discussed in the introduction). It is a qualitative classification apparently achieved by human eye using molecular graphics software. Our aim here is to develop a quantitative method for assignment of hinge and shear that would correspond to the qualitative assignment by Gerstein et al. As both the DBMM and the NRDB are based on the PDB, PDB files can be used as the point of reference to make a direct like-for-like association.

The NRDB is organised by protein family, where PDB entries are collected and domain movements grouped by a conformational clustering method [97]. If the conformational clusters are "tight" the two best (based on resolution) representative structures are analysed by DynDom and the domain movement represents the domain movement between all pairs from the two clusters. From the representative PDB code, the PDB codes of all structures within the cluster can be found. Some clusters are not tight but "extended". For these, all possible pairs are considered but not all of them are used as input to DynDom as many of the movements they imply can be represented by movements between particular pairs using a dimensional clustering method. A match between the NRDB2d and DBMM was assigned if either of the DBMM structures were found in different conformational clusters in the NRDB, or both DBMM structures were from the same "extended" cluster in the NRDB. The DBMM was data-mined using "web scraping" techniques [139] which is a program that dynamically navigates through a series of webpages, collecting specified data within it based on HTML coding. This process allowed us to make an association between pairs of structures in the NRDB and the DBMM through their PDB codes. It allowed us to evaluate the N value for each domain movement (using the DynDom analysis and our contact analysis described above) and consider this value against its DBMM assignment of hinge and shear.

4.4 ROC Analysis of N Value for Hinge & Shear

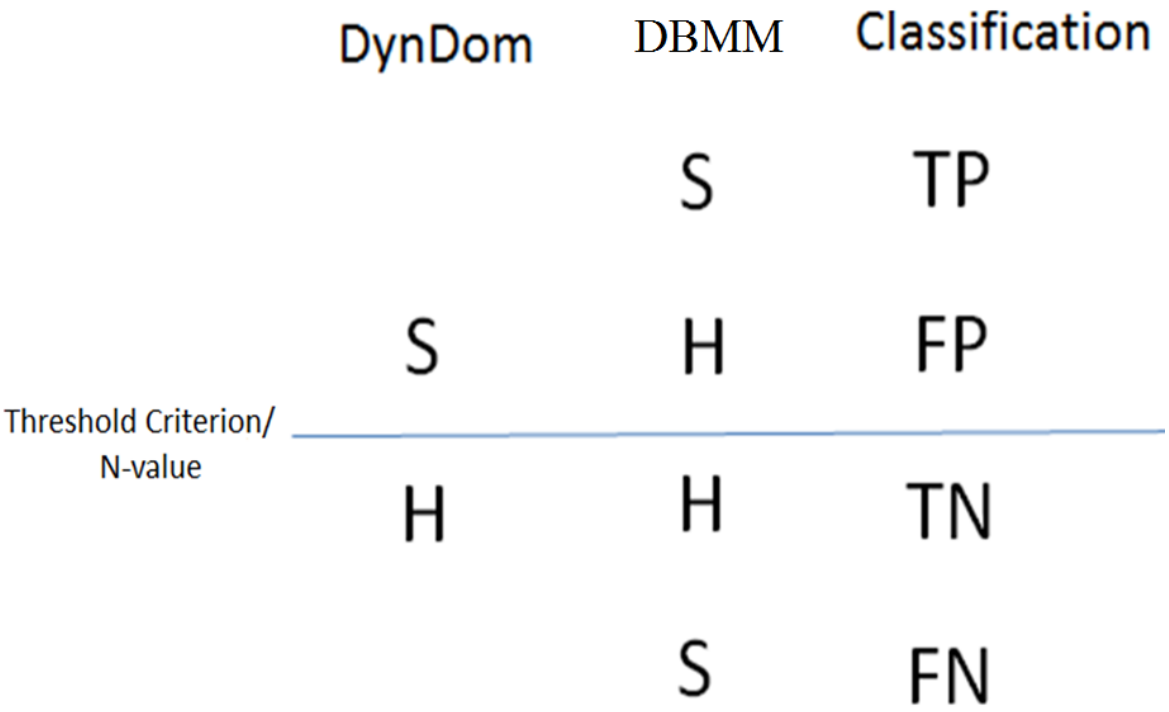

Fig. 4.8 DynDom vs. DBMM ROC classification cut-off.

The DBMM assignments are considered to be the gold standard against which the N-value predictions are compared. If we consider a DBMM shear as an actual positive and a DBMM hinge as an actual negative and we apply a threshold to our N value such that a N value above this threshold is a prediction for shear and a N-value below it is a prediction for hinge, then we can count the number of True Positives (shear, correctly predicted shear), False Negatives (shear, incorrectly predicted hinge), True Negatives (hinge, correctly prediction hinge) and False Positives (hinge, incorrectly predicted shear) (Figure 4.8) and, as described in the Methodology section, plot a ROC curve by varying the threshold. The ROC curve produced shows reasonable predictive power in the N value. The area under this ROC curve is 0.78 showing this method is a good predictor for hinge and shear (Figure 4.9). This ROC curve was based on (Equation 4.24) for the N value but there was no “learning” or “training” involved. In the next section we describe the result of applying logistic regression applied to

this data.

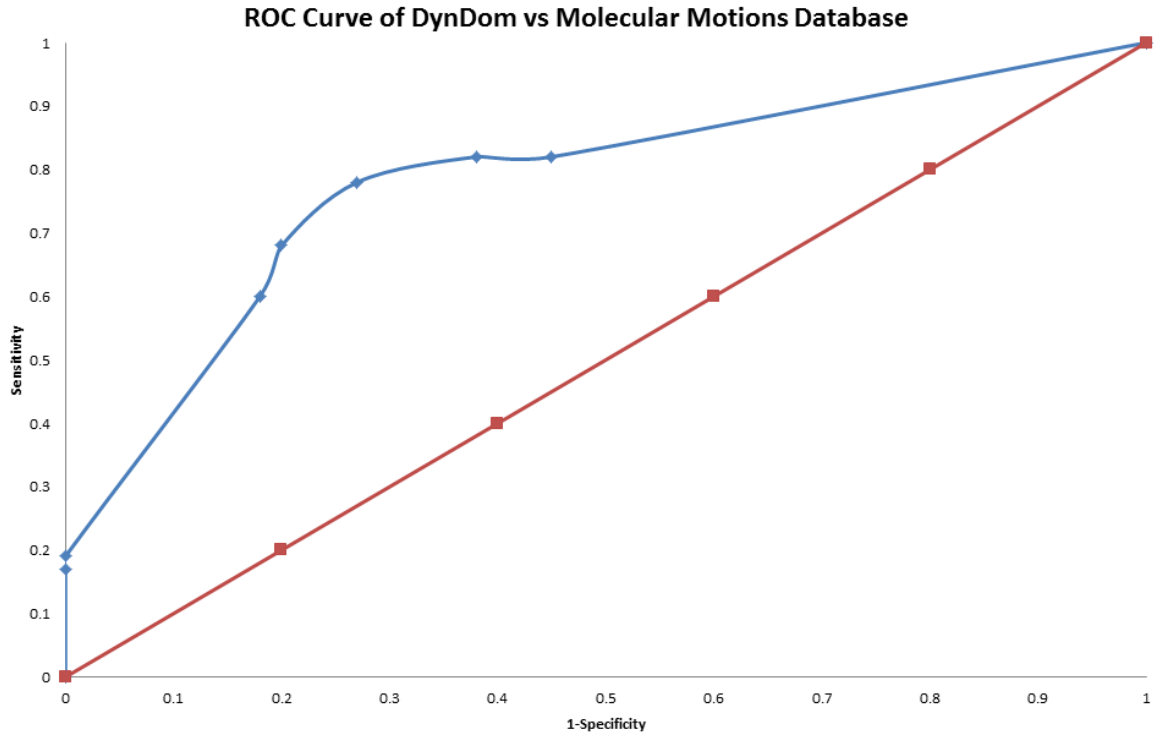

Fig. 4.9 Preliminary DynDom vs. DBMM ROC Curve Results.

4.5 Domain Motion: Logistic Regression

Using the logistic regression calculations (subsection 3.3.3): When matching cases between the DBMM and NRDB2d datasets 37 "predominantly shear" or shear domain movements were found in the DBMM, of which 21 were found in NRDB2d. 75 "predominantly hinge" or hinge domain movements in the DBMM, with 41 found in NRDB2d. In addition to the examples found in the NRDB2d, the DBMM produced 2 further cases in the shear case and 13 in the hinge category, on which DynDom was run directly, producing in total 77. The N_p and N_u values for all 77 domain movements were computed and logistic regression performed and applied to all to the whole of the NRDB2d dataset (as described in the Methodology section). The following logistic equation was found (Equation 4.25):

$$y(N_p, N_u) = \frac{1}{1 + e^{(-0.147N_p + 0.1994N_u - 0.0423)}} \tag{4.25}$$

To test the predictive power of this model against DBMM assignments a new ROC curve was produced. Like before (Figure 4.10) a positive expectation is a prediction for "shear" when $y(N_p, N_u) > y_{cutoff}$ where y_{cutoff} is selected in 0.1 intervals between 0.0 and 1.0. The area beneath the ROC curve is calculated as 0.83 indicating that this model is a good predictor of hinge and shear movements.

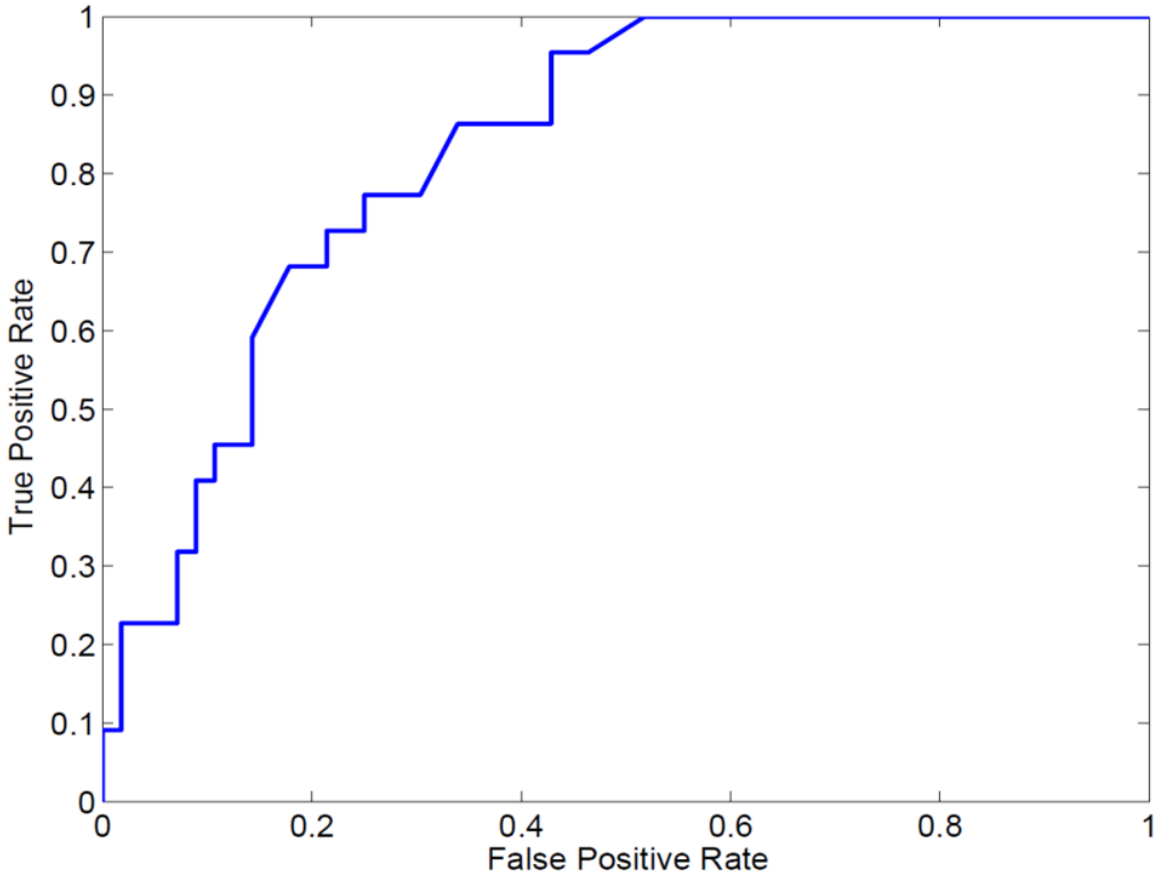

Fig. 4.10 Logistic Regression ROC Curve.

To further clarify this result, a leave-one-out cross-validation method was used (Figure 4.11). The area under this ROC curve is 0.78 endorsing that the logistic regression technique (Equation 4.25) is a good predictor of hinge and shear.

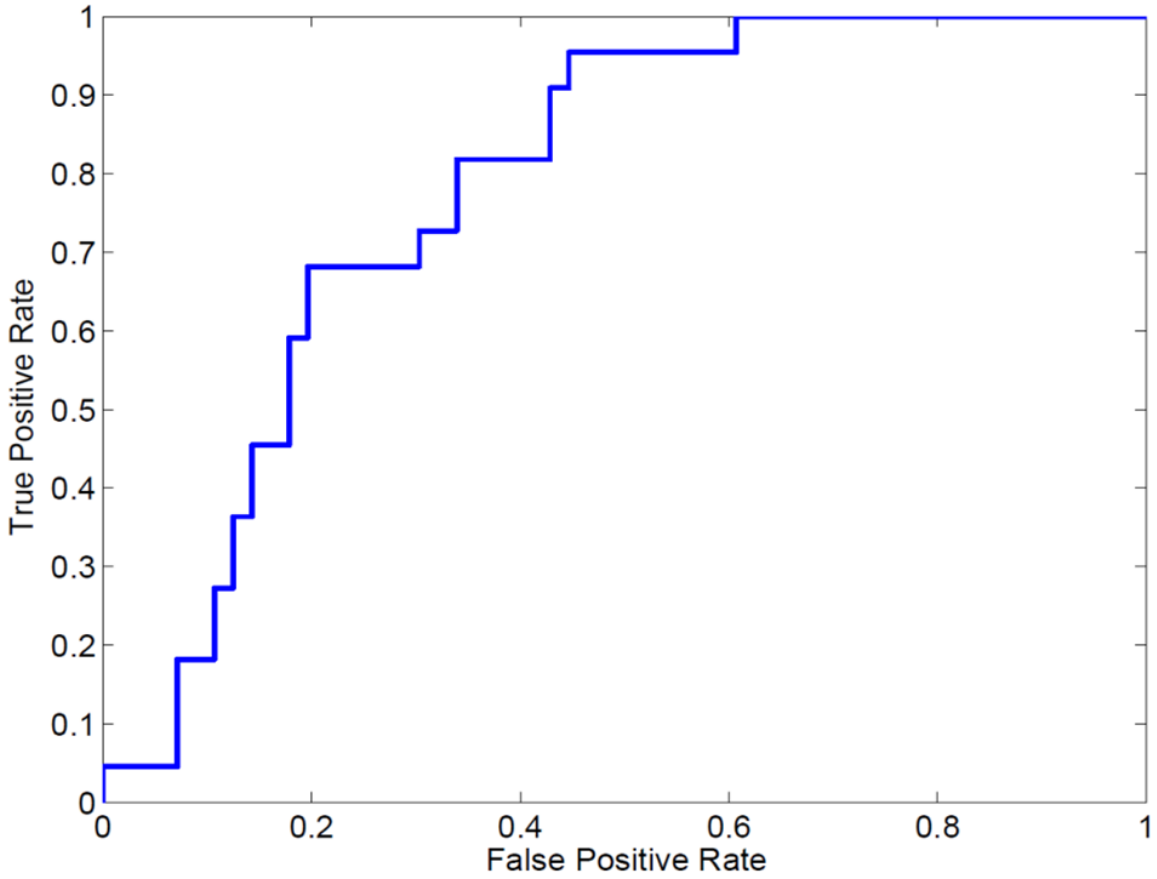

Fig. 4.11 Leave One Out ROC Curve Graph.

If domain movements belong to two well-defined classifications, a plot of the distribution of the $y(N_p, N_u)$ might indicate clustering. Conversely, a binned histogram of the $y(N_p, N_u)$ frequencies of for all 1822 in NRDB2d presented a flat distribution. The accumulative frequency distribution when plotted was linear. Because of the lack of evident clustering, the four classes were created (Equation 4.26):

$$\begin{aligned}
&\text{Strong Hinge} = 0 \leq y(N_p, N_u) < 0.25 \\
&\text{Weak Hinge} = 0.25 \leq y(N_p, N_u) < 0.5 \\
&\text{Weak Shear} = 0.5 \leq y(N_p, N_u) < 0.75 \\
&\text{Strong Shear} = 0.75 \leq y(N_p, N_u) \leq 1.0
\end{aligned} \tag{4.26}$$

The precision of a classification can be determined as the percentage of correctly predicted DBMM cases to be in that class to the total number of DBMM cases predicted to be in that class. 39 DBMM cases were predicted to be strong hinge, of which 35 were hinge according to DBMM giving a precision of 90%, indicating a good relationship between predicted hinge and a "true" hinge allocated by DBMM. The precision of strong shear has only 3 DBMM cases being predicted to be strong shear (of which 2 were shear giving a precision of 66.7%).

4.6 Categorisation based on N_p, N_{u1} and N_{u2} interdomain contacts classifications and limitations

Hinge and shear mechanism seems to be a rather coarse level of categorisation. It would appear that this might be improved upon by extending the residue contact change methodology developed here. If we consider just the absence or presence of contacts between two domains in a domain movement there are three categories of domain movements: noncontact-to-noncontact, contact-to-noncontact (same as non-contact-to-contact), and contact-to-contact. By considering at most a single pairwise contact between the domains in either conformation these three categories can be further subdivided into five fundamental "contact-change" scenarios. The contact-changes can be associated with five "model" domain movements assuming the following idealised scenario. Firstly domains have a spherical shape; secondly there is only one residue from each domain at a contact point; the relative movement of the domains is a rotation about a hinge axis passing through an interdomain linker region. These elemental contact changes with the model domain movements are based on the simplest domain movement to replicate the elemental contact-change in an idealised system. The extent to which real domain movements conform to these idealised movements is something to be determined.

4.6.1 Noncontact-to-noncontact (Null-to-Null)

$$\begin{aligned} N_p &= 0 \\ N_{u1} &= 0 \\ N_{u2} &= 0 \end{aligned} \tag{4.27}$$

This noncontact-to-noncontact set is acknowledged in (Equation 4.27) with no contacts in both conformations. An overwhelming majority have a single linker. A typical example is shown in (Figure 4.12). This "null-to-null" case indicates the domains stay separated and

move freely.

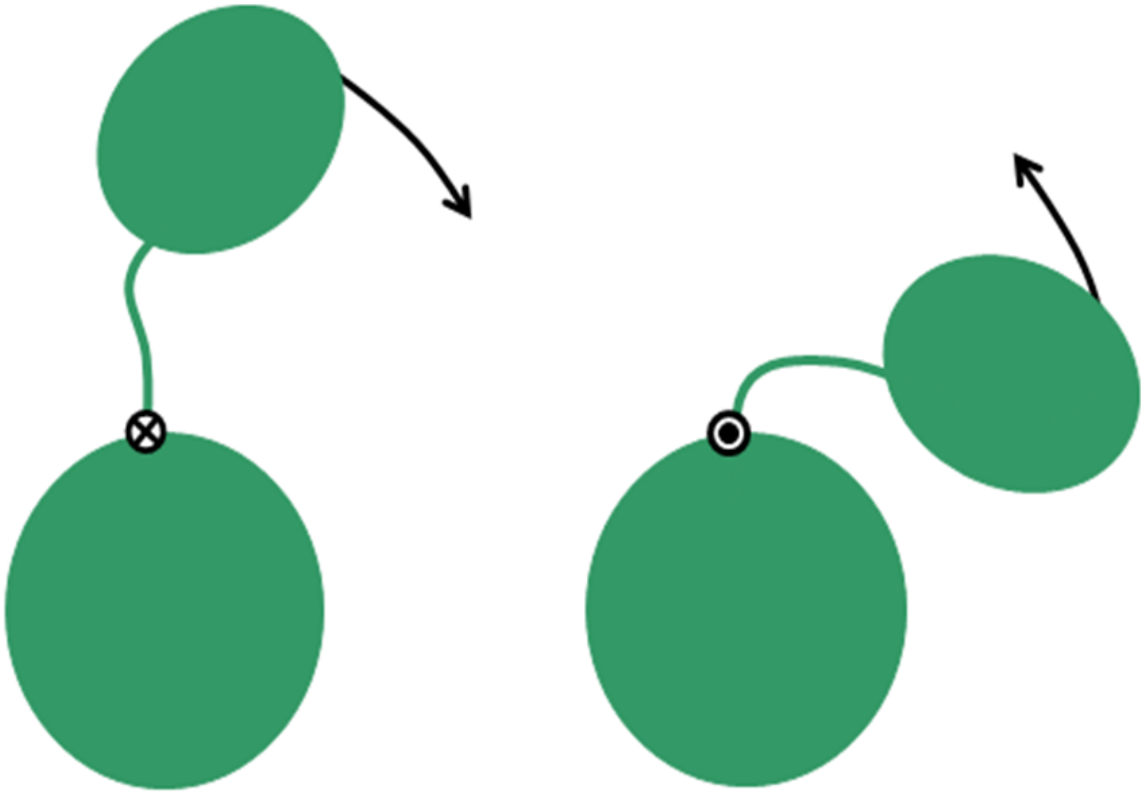

Fig. 4.12 schematic illustration of noncontact-to-noncontact (Null-to-Null) creating the "free" movement. (Black X indicates conformation 1 and black dot indicates conformation 2)

4.6.2 Contact-to-noncontact (New)

$$\begin{aligned} &N_p = 0 \\ &N_{u1} \neq 0 \\ &N_{u2} = 0 \\ &or \\ &N_p = 0 \\ &N_{u1} = 0 \\ &N_{u2} \neq 0 \end{aligned} \tag{4.28}$$

This group is identified by (Equation 4.28). It has no preserved contacts and only has

contacts in one conformation and no contacts in the other; this could be conformation 1 or 2. This kind of domain movement can be thought of as an "open-closed movement" and can be imagined as a door opening and closing mechanism (Figure 4.13). This suggests a rotation about a hinge axis perpendicular to the line connecting the centers of mass of the domains, defined formerly as a “closure” motion [49].

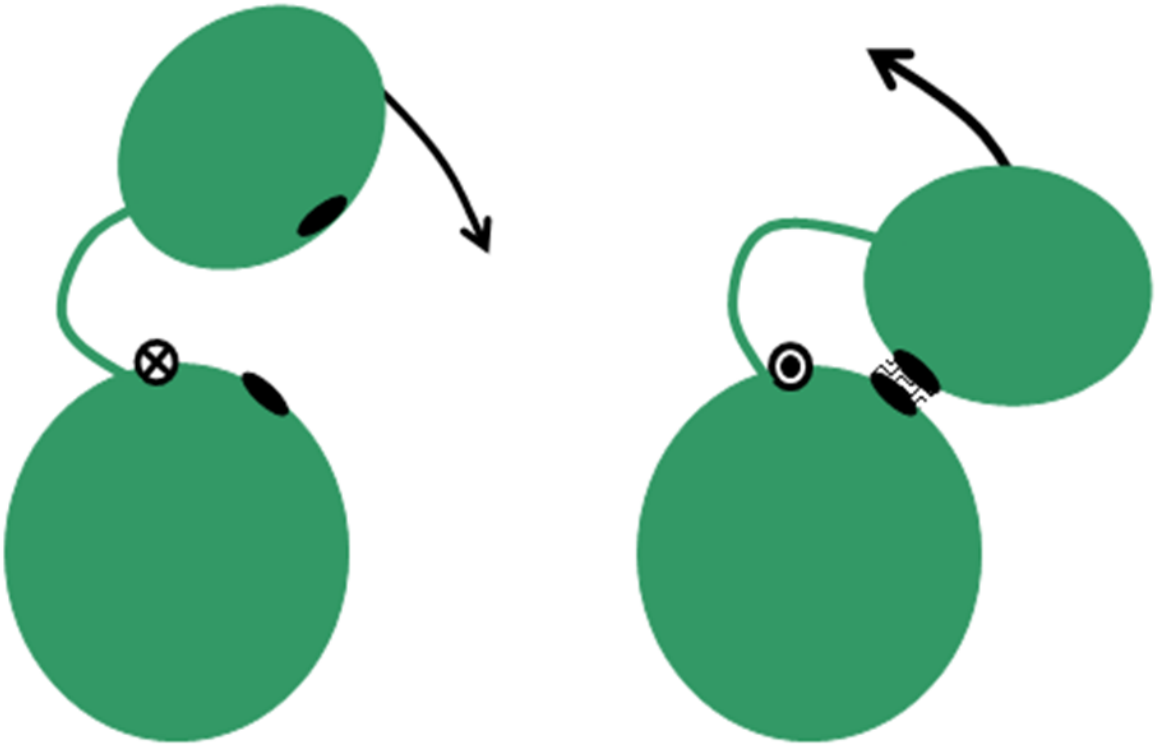

Fig. 4.13 Schematic illustration of contact-to-noncontact (New) creating the “open-closed” movement.

4.6.3 Contact-to-contact

This category can be further subdivided into three further subcategories because of the different combinations of unique to preserved contacts.

Maintained:

$$\begin{aligned} N_p &\neq 0 \\ N_{u1} &= 0 \\ N_{u2} &= 0 \end{aligned} \tag{4.29}$$

Maintained is defined by set theory as in (Equation 4.29). This group has preserved contacts and would appear to represent the perfect examples of shear in that all residue contacts between the domains are preserved as would be the case for "interdigitating sidechains". Its motion can be viewed as (Figure 4.14), indicating the domains cannot move (given the exclusion of the hinge region) implying the domains remain "anchored".

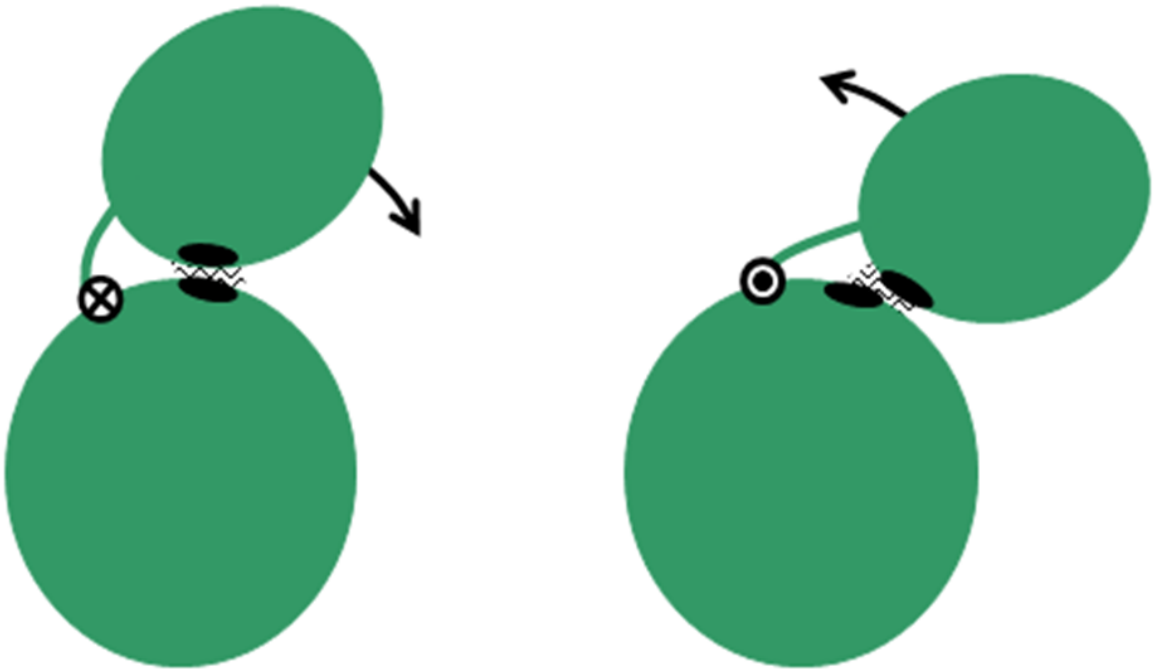

Fig. 4.14 Schematic illustration of contact-to-contact (Maintained) creating the "anchored" movement.

Exchanged-Pair:

$$\begin{aligned} N_p &= 0 \\ N_{u1} &\neq 0 \\ N_{u2} &\neq 0 \end{aligned} \tag{4.30}$$

Exchanged-Pair has no preserved contacts, only new contacts established in both conformations. It can be defined by set theory as (Equation 4.30). The domain movement swings from a "closed-on-one-side" to "closed-on-the-other-side" conformation, which in a study of Lactoferrin has previously been referred to as a "see-saw" movement [40]. Lactoferrin is found amongst this group. This group comprises examples that have very large angles of rotation (Figure 4.15). The two residues making contact in one conformation are not involved in making contact in the other conformation, suggesting a movement with the hinge axis perpendicular to the line connecting the centers of mass, breaking contact on one side of the domains and rotating until contact is made on the other side of the domains.

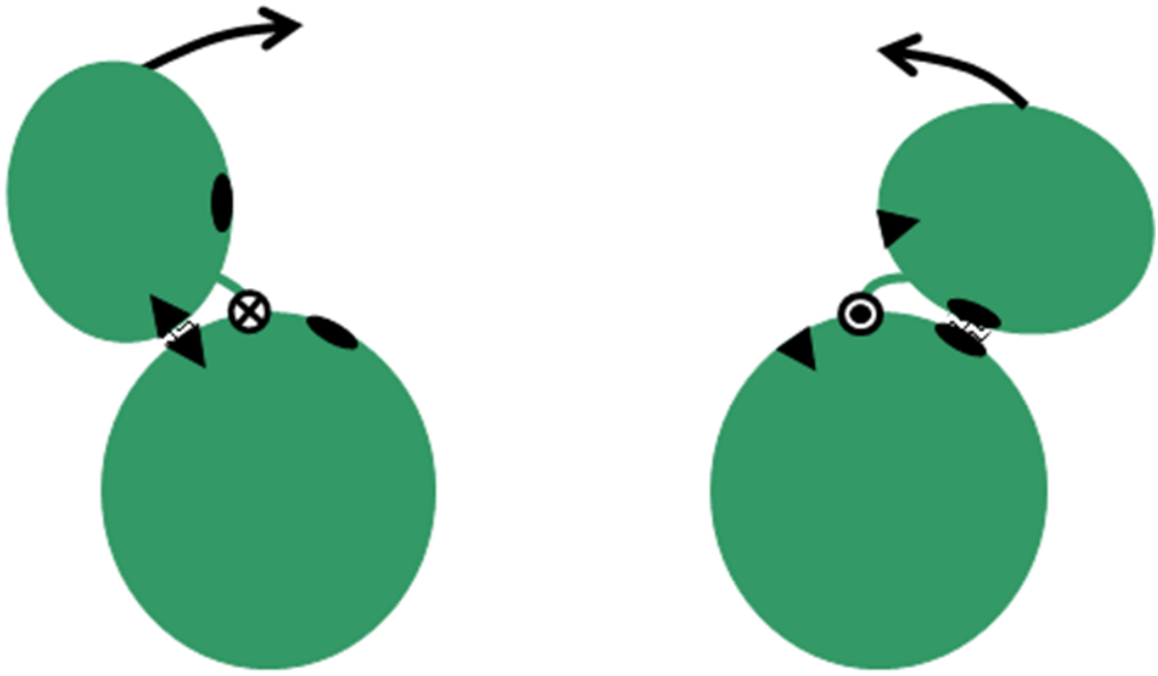

Fig. 4.15 Schematic illustration of contact-to-contact (Exchanged-Pair) creating the "see-saw" movement.

Exchanged-Partner:

$$\begin{aligned} N_p &\neq 0 \\ N_{u1} &\neq 0 \\ N_{u2} &\neq 0 \end{aligned} \tag{4.31}$$

Analysis of all movements in contact-to-contact is made difficult by definition of "preserved contacts", which can mean the residue maintains contact with the same residue in the other domain between the two conformations, or, that the residue exchanges contact with another residue, thus the term "exchanged-partner". The latter would arise when the domains slide over each other and is therefore particularly relevant to this study as it suggests a sort of "shear movement." This can be defined according to set theory as (Equation 4.31). If the movement is controlled by a rotation about a well-defined hinge site, it would be clearer to refer to these as "sliding-twist" movement (Figure 4.16) indicating one domain sliding over the other by a relative twist of the domains. In this mechanism the hinge axis passes through the center of mass of domain A, with the center of mass of domain B slightly shifted from the hinge axis, giving a twist motion. If contact occurs between the two domains then the contact point (a residue on domain B) will follow a circle on domain A (so residue B will contact two different points on domain A in a movement). This can be seen as a "sliding-twist" motion.

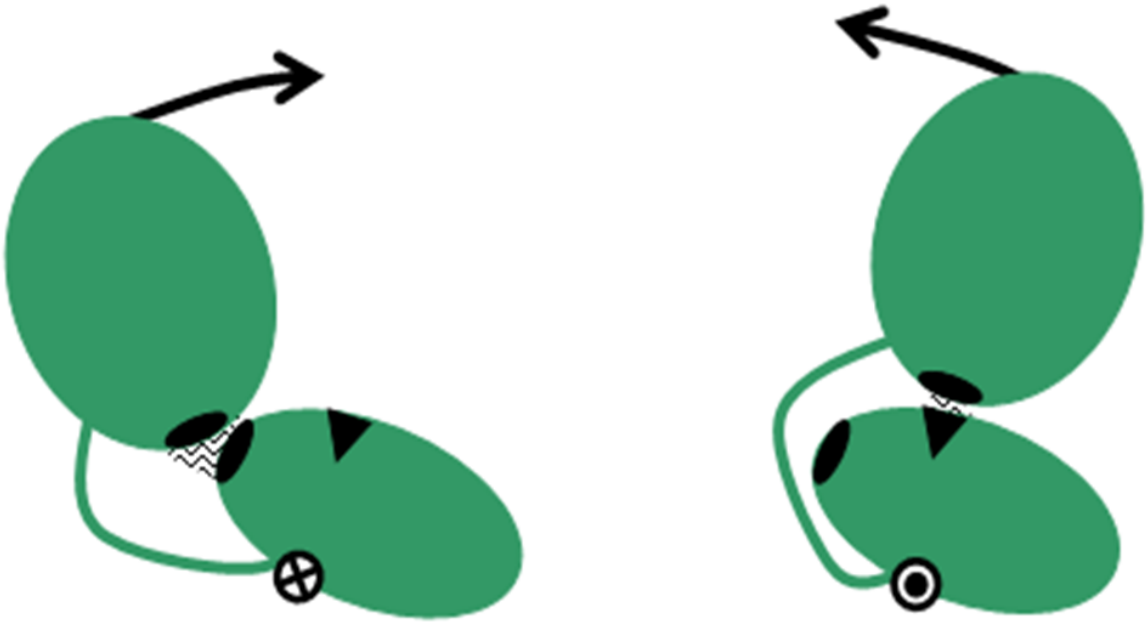

Fig. 4.16 Schematic illustration of contact-to-contact (Exchanged-Partner) creating a "sliding-twist" movement.

The complexity of the contact-to-contact subclass means that distinguishing between the three groups, maintained, exchanged-pair and exchanged-partner, cannot be based purely on the unique and preserved contact analysis as described above. Further analysis is needed to identify which specific movement has taken place.

4.7 Further Analysis of Contact-to-Contact Class

The set theory analysis to identify preserved or unique contact changes between the two conformations is limited for the analysis of contact-to-contact set as it would be unable to distinguish Maintained, Exchanged-Partner and Exchanged-Pair contact changes except when there is only one contact made per conformation. To remedy this, a different approach is required.

4.7.1 Contact Pairs

The problem with our approach is that we record and classify individual residues according to whether they make contact or not in both conformations but we omit to record information on the residues they contact. In order to remedy this we record the pairs of contacting residues for each conformation. From the basic DynDom output file (RasMol script file) we create lists of pairwise residue contacts as shown in (Table 4.1). Each conformation has its own column and within that column the domain A residue number on the left makes a contact with domain B residue number on the right separated by a "=" indicating contact. The number indicates the residue number in that PDB file. Using this example the contact made between amino acid 72 in domain A and amino acid 49 in domain B is preserved between the dynamic movement, but residue 49 in domain B also makes contact with residue 70 in domain A in the first conformation and likewise residue 70 only makes contact with residue 66 in the second conformation. This interpretation of the data already gives an instant and more detailed understanding of the movement taking place. There is a maintained contact but also at least one exchanged-partner contact being made.

AUTOLYSIN	
1GVM(F)	2BML(B)
72=49	72=49
70=49	70=66
70=58	

Table 4.1 Example of a contact pair table, with DynDom ID at the top followed by two columns, at the top of each column there is the PDB ID and in () the chain ID followed by the residue numbers of opposite domains making contact with one another.

This further analysis compares the residues between both conformations, particularly to see if the same individual residues or contact pairs are preserved in both conformations. If this contact pair data could be represented by a graph then it would not only provide us with something visual but it would also allow it to be analysed using the many tools developed from graph theory.

Chapter 5

Results: Dynamic Contact Graphs

5.1 Dynamic Contact Graph Introduction

A new analysis developed as part of this research is presented as Dynamic Contact Graphs (DCGs) which provide the answer to contact pair data classification. Let $\{(a_{1i}, b_{1i})\}, i = 1.....N_1$ denote the set of residue contact pairs in conformation 1 and $\{(a_{2i}, b_{2i})\}, i = 1.....N_2$ the corresponding set for conformation 2. Each node of the graph represents a residue of which there are two types: those in domain A and those in domain B. An edge exists when there is a contact between a residue in domain A and a residue in domain B, i.e. when they appear in one of the sets above. The key feature of a DCG is that it is a directed graph. For contacts in conformation 1 the direction associated with an edge is from the residue (node) in domain A to the residue (node) in domain B (call this an AB edge). This could be written as $a_{1i} \rightarrow b_{1i}$. For contacts in conformation 2 the direction is from the residue (node) in domain B to the residue (node) in domain A (call this a BA edge). This could be written as $a_{2i} \leftarrow b_{2i}$

(Equation 5.1).

$$\begin{aligned} &\text{Conformer 1 Contact: Residue A} \longrightarrow \text{Residue B} \\ &\text{Conformer 2 Contact: Residue B} \longrightarrow \text{Residue A} \end{aligned} \tag{5.1}$$

In general a domain movement may combine various contact-changes and have a complex graph structure. We make full use of Matlab (version 8.0.0.783 (R2012b)) and in particular the Bioinformatics Toolbox "biograph" function to create a "biograph" object, a data structure for directed graphs. This enabled us to use associated methods to analyse and view the DCGs.

5.1.1 Elemental Contact Change: Null-to-Null

The noncontact-to-noncontact free-linker category by its very definition has no contacts in either of the conformations and so the DCG is an empty graph.

5.1.2 Elemental Contact Change: New

In the contact-to-noncontact group, which is the New case there is a single arrow the direction of which indicates the conformation in which the contact takes place (Figure 5.1).

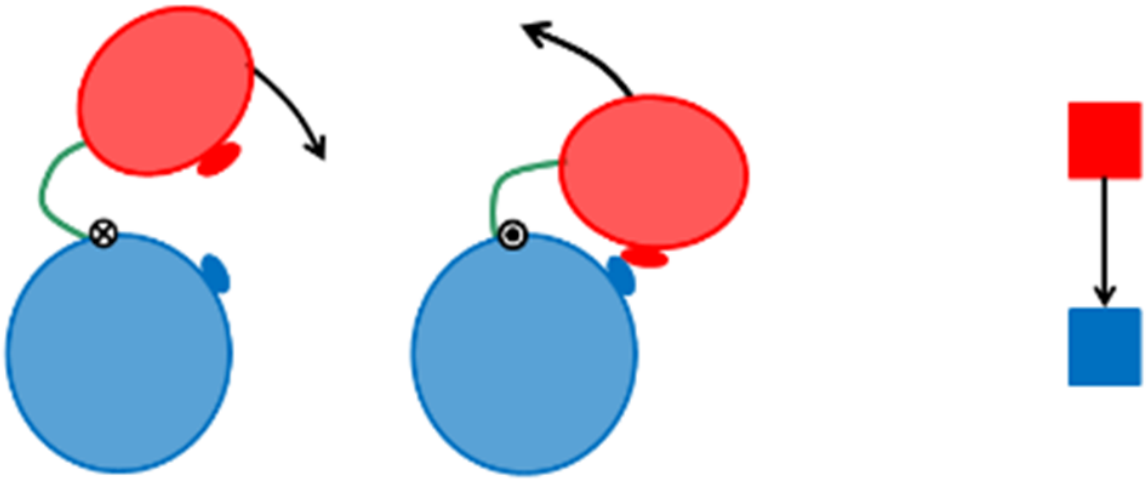

Fig. 5.1 DCG representation of a New Motion.

5.1.3 Elemental Contact Change: Maintained

A Maintained contact gives two directed edges (arrows) connecting them (Figure 5.2), indicating that this contact pair exists in both conformations.

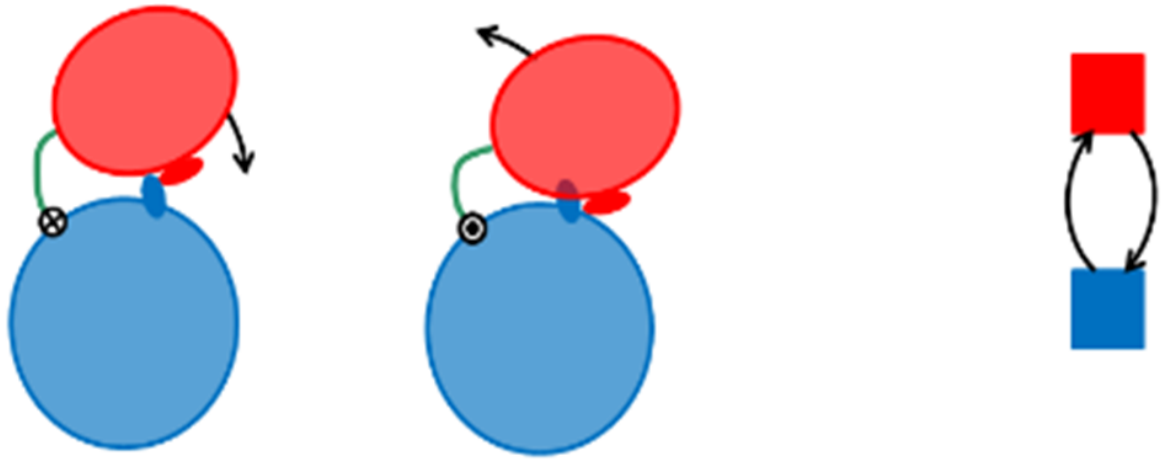

Fig. 5.2 DCG representation of a Maintained Motion.

5.1.4 Elemental Contact Change: Exchanged-Pair

The Exchanged-Pair movement in many ways looks very similar to the New (Figure 5.3). However, unlike New there will be arrows in both directions but disconnected from one another because the unique contacts are made in each conformation.

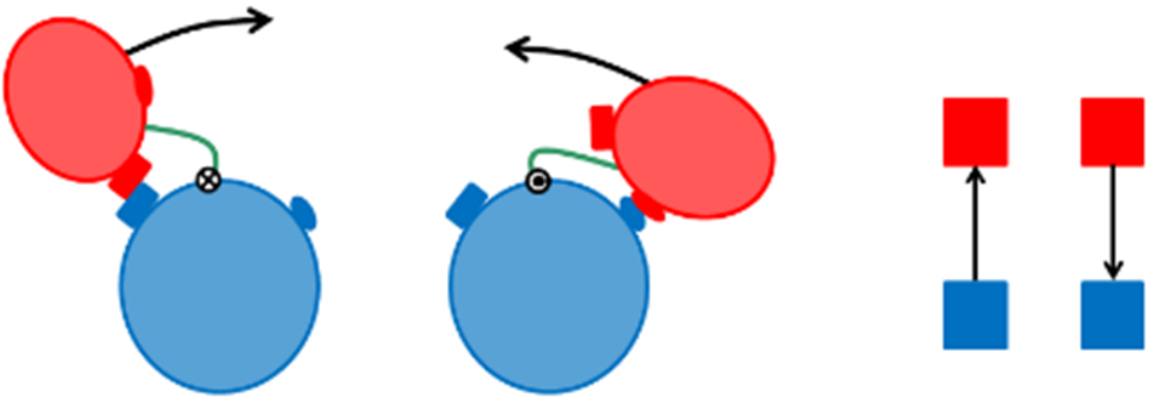

Fig. 5.3 DCG representation of an Exchanged-Pair Motion.

5.1.5 Elemental Contact Change: Exchanged-Partner

The Exchanged-Partner (Figure 5.4) case is perhaps the most complex to decipher from the DCG's as it is both a preserved contact and a unique contact in our original single residue contact based analysis. A sliding motion can be observed if there is a line/sequence of alternating contact partners of contacter-contactee-contacter or contactee-contacter-contactee, or if viewed on the DCG red-blue-red or blue-red-blue. This indicates the residue in the middle is making contact with both residues either side, one in each conformation.

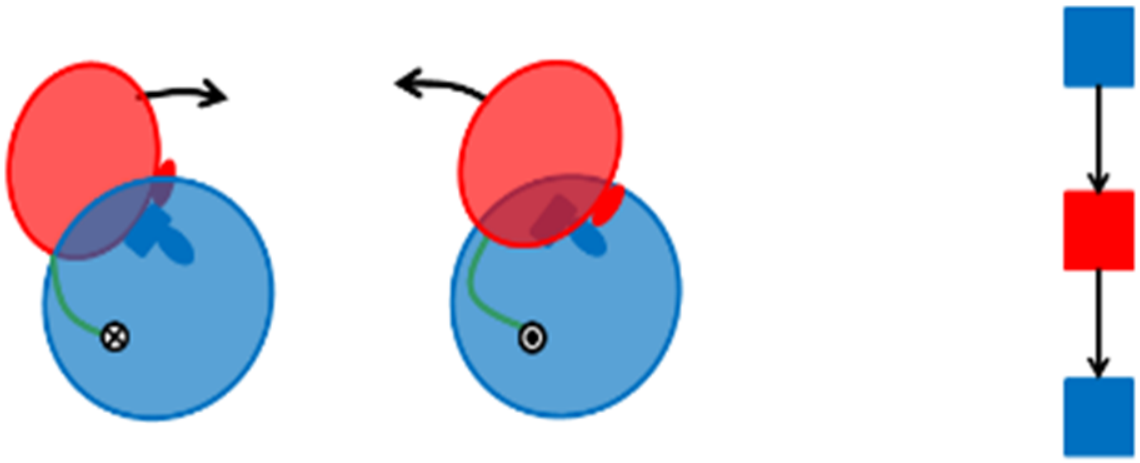

Fig. 5.4 DCG representation of a Exchanged-Partner Motion.

5.2 DCG Classification of Protein Domain Movements

Each dynamic pair movement in the NRDB2d can be represented with a DCG (Figure 5.5). In all 413 domain movements have no contacts in both conformations and come under the null-to-null definition (ignoring the contacts removed from the interdomain bending region and within 5.5Å of the axis). These domain movements are allocated to “pure no contact” class, implying freedom of movement between the domains.

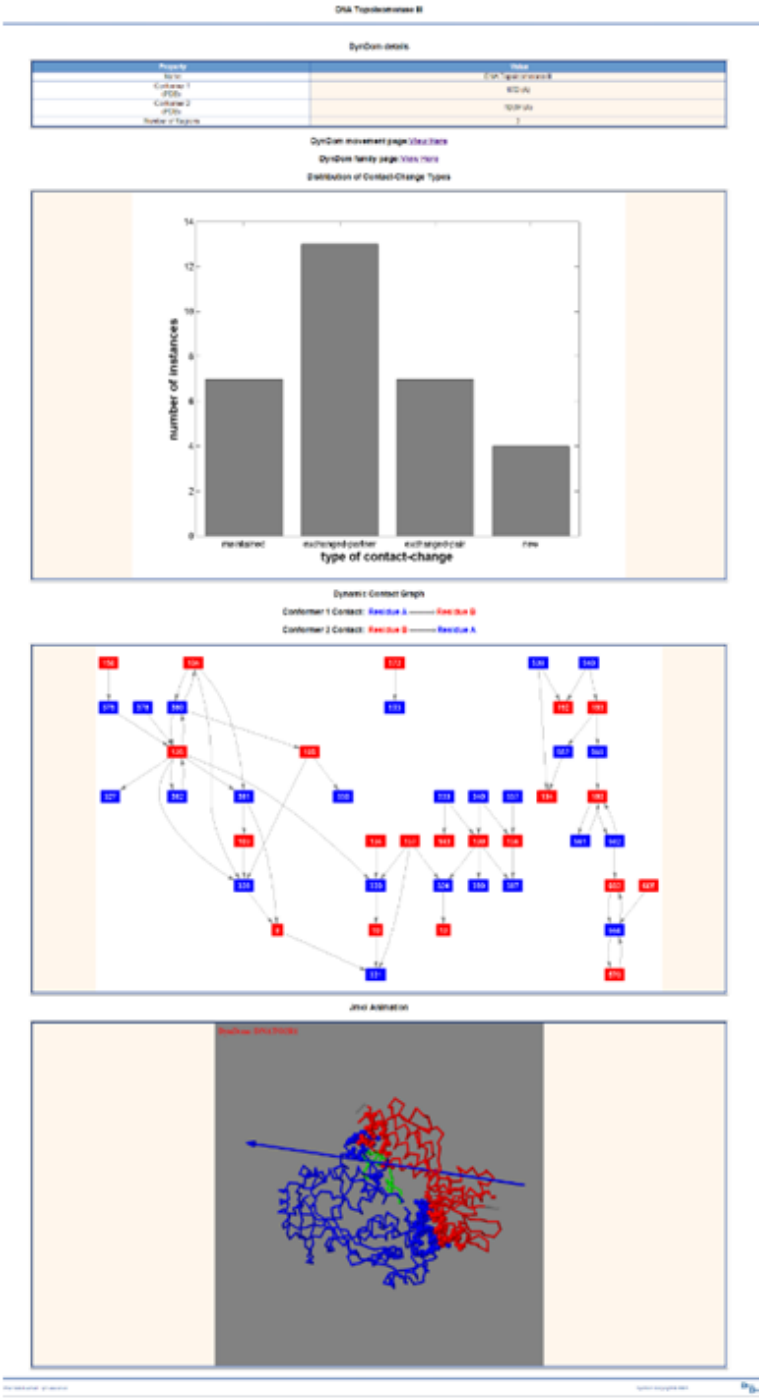

Fig. 5.5 Example of a dynamic pair individual website (DNA Topoisomerase III) which includes name of the protein, the 2 PDB code ID's with corresponding chain identifiers, number of connected/disconnected regions, bar chart to indicate number of pairwise interatomic contacts.

The DCG can be interpreted by eye to give an indication of the kinds of movements

occurring. (Figure 5.6) highlights one such example, where a maintained contact can be observed (241=264) and there is one sliding residue, 262, which exchanges partner 250 with 258 or, 241 with 258.

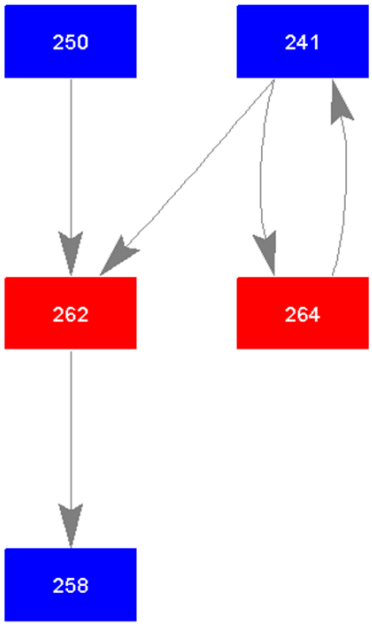

Fig. 5.6 Example of the DCG graph in Autolysin.

5.2.1 Deconstructing DCGs into the contact-changes classes:

As can be seen in (Figure 5.5) DCGs can be complicated and are not always connected. A disconnected graph means that a set of contacting residues in one subgraph make no contacts with the set of residues in another subgraph. Residues in these disconnected subgraphs are likely to represent remote regions that play a different role in the domain closure process. It is of interest to evaluate the number of these regions by counting the number of disconnected subgraphs. This can be measured by using the Matlab Bioinformatics Toolbox's "biograph" object method "conncomp" to count the number of disconnected subgraphs in the DCGs (see Methods section). In order to classify domain movements the DCG's can be deconstructed into their four elemental contact-change categories (maintained, new, exchanged-pair and exchanged-partner). Doing this will allow us to count the number of elemental contact

changes in each of the four classes and then classify the domain movements. Identifying "maintained" contact-changes (Figure 5.2) is very easy, because the same contact pair will exist in both conformations, however counting "exchanged-partner" contact-changes (Figure 5.4) is non-trivial as illustrated in the example shown in (Figure 5.7).

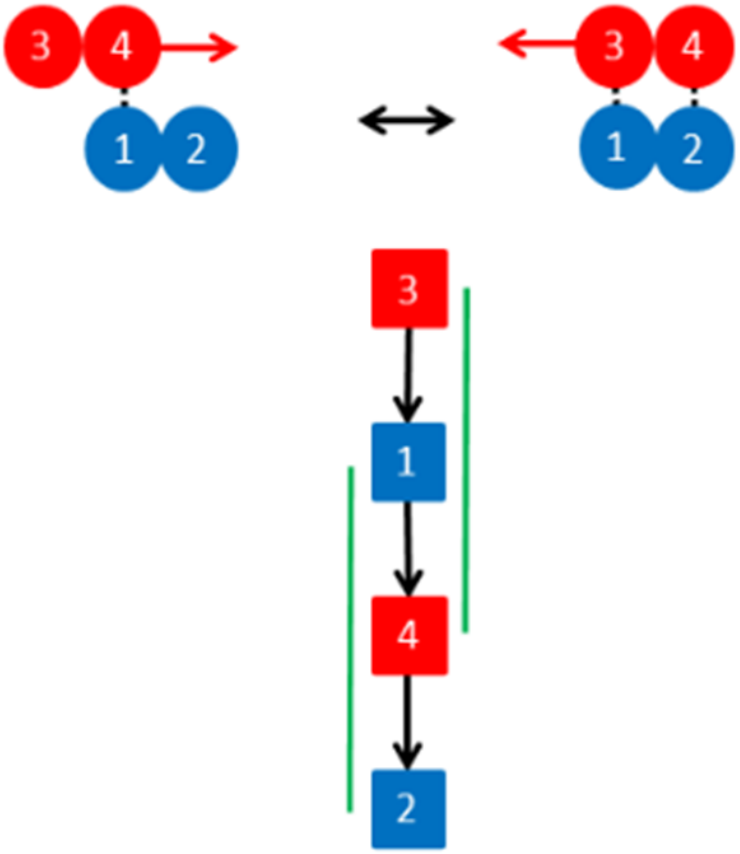

Fig. 5.7 DCG example of Exchanged Partner contact change.

The problem comes from a perspective on how the motion is viewed. Is residue 1 making contact with residue 4 then sliding across onto residue 3, or is residue 4 making contact with 1 before sliding across onto residue 2, in other words, which residue is making the exchange? In the DCG for this set of contact changes this maps into the issue of identifying triplets (a row of 3 nodes connected by 2 co-directed edges which is the elemental DCG for an exchanged-partner contact change). In this example there are two triplets, 3-1-4 or 1-4-2, but counting them both would mean counting the 1-4 twice. Therefore only one of

these triplets is selected. This means generally we should not count overlapping triplets but non-overlapping triplets to determine the number of exchanged-partner contact changes.

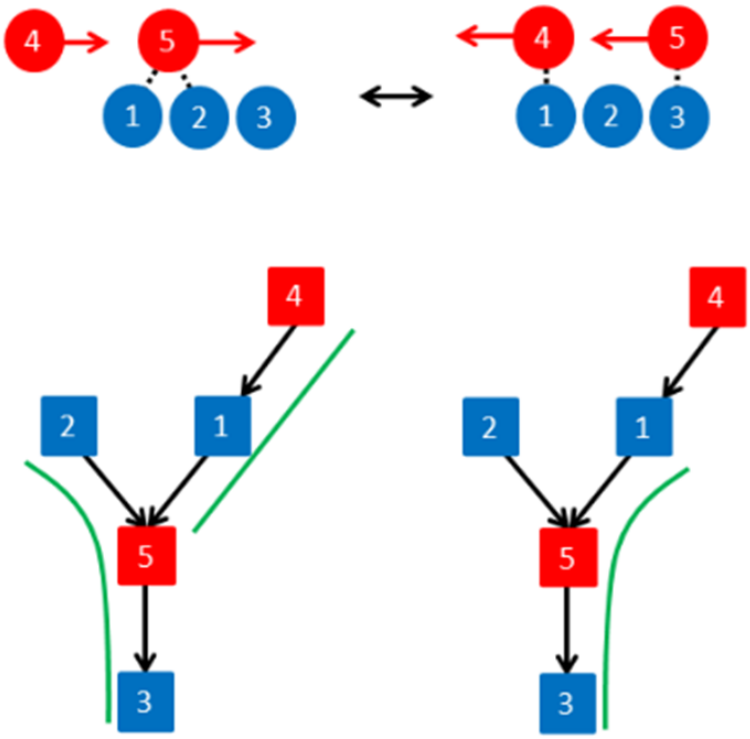

Fig. 5.8 Exchange Partner interaction with sliding contacts to multiple partners.

(Figure 5.8) presents an example with two possible solutions. (1) Two "exchanged-partner" contact-changes: residue 1 (in domain A) sliding from residue 5 (in domain B) to residues 4 (in domain B), and residue 5 (in domain B) sliding over to residues 2 and 3 (in domain A). (2) just one "exchanged-partner" contact-change: residue 5 (in domain B) makes contact with residue 3 (in domain A) and then slides over to residue 1 (in domain A). This would give the interactions between residues 2 and 5, and 1 and 4 in conformation 1 and conformation 2 respectively, assigning them "exchanged-pair" contact change. Thus there are two ways to accommodate non-overlapping triplets in this example, one giving one triplet, the other giving two triplets. Although both are possible, it is more likely that, given some of the residues are in an exchange-partner contact-change indicating a sliding movement, then all residues would be sliding and therefore in an exchanged-partner contact change.

Therefore we should maximise the number of exchanged-partner contact-changes in a graph. An alternative argument would be that we should maximise the number of associated contact pairings in a graph (in an exchanged-partner contact change two contact pairs one from each conformation are associated via the residue that appears in both) before pairing off contact pairs to the exchanged-pair contact-changes for which there is no association.

5.2.2 Algorithms for DCG deconstruction for contact-change classification

The maintained contact-changes are the first to be identified in the DCG, which in turn creates a different DCG with no doubling linked nodes. The number of non-overlapping triplets in the resulting graph can then be determined. Firstly all overlapping and non-overlapping triplets are found. Then a new (undirected) graph is created with each triplet represented by a node (vertex) and an edge joining any two nodes from triplets that overlap. This is then searched using a branch and bound routine to determine the largest number of non-overlapping triplets. The algorithm involves selecting a node, removing those nodes connected with it by a single edge and repeating this process until no nodes remain. The selected nodes give a set of non-overlapping triplets. This recursive program is given here in pseudo-code (Figure 5.9):

```
Input: A graph with vertices (nodes, representing
triplets) ordered, V=v1,v2,v3,.. ,vn and a set of edges E
(an edge existing if the two vertices represent triplets
that overlap).

Output: A list of vertices, Wmax, with the maximum number
of vertices, Nmax, none of which are connected by a single
edge.

Nmax=0
Wmax={}
W={}
add v1 to W
w=v1
V'=V
unconnected(w,V',E,W,Wmax,Nmax){
      if (|V'|=0){
            if (|W|>Nmax){
                  Wmax=W
                  Nmax=|W|
            }
            return Wmax,Nmax
      # terminate branch in search tree if it cannot
      # exceed Nmax
      }elseif (|V'|+|W|<= Nmax){
            return
      }
      while (there is an edge (w,vj)∈E) {
            remove vj from V'
      }
      remove w from V'
      add vi to W #vi appears first in V'
      w=vi
      # recursive call to unconnected
      unconnected(w,V',E,W,Wmax,Nmax)
```

Fig. 5.9 Pseudo-Code for DCG analysis of constituent contact changes.

The size and complexity of the 12 DCGs proved problematic for this algorithm in terms of CPU time, and so a different algorithm which used a related random search was used instead (Figure 5.10).

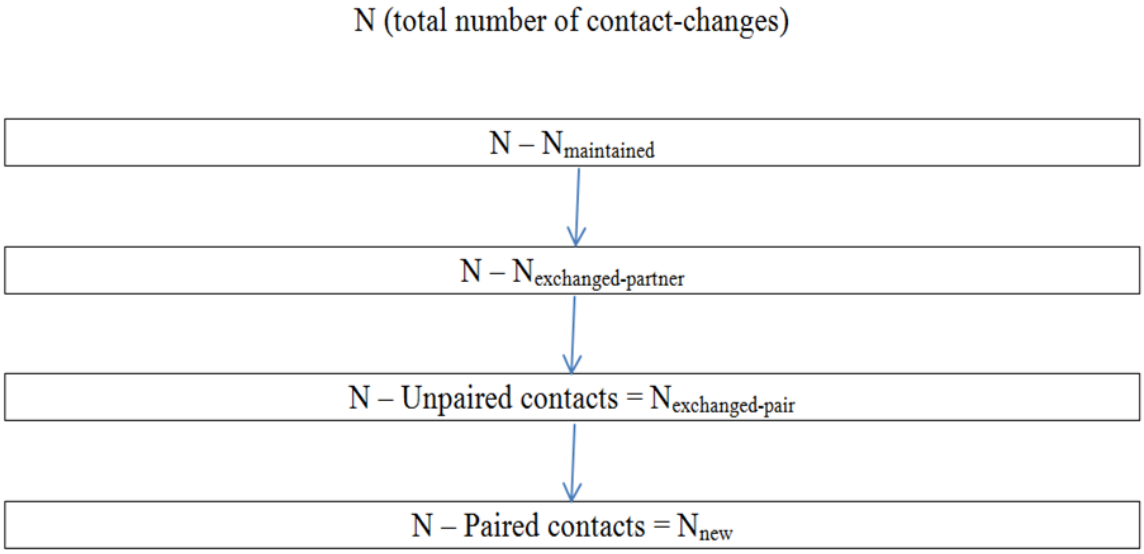

Fig. 5.10 Random Related Search algorithm alternative for CPU intensive DCG's.

This algorithm found the same value of N_{max} as the main searched algorithm in all 1397 DCGs that could be search using the main algorithm. The resulting maximum number of non-overlapping triplets does not correspond to a unique set, but this is of no consequence here as we are only interested in the maximum number of exchanged-partner contact-changes.

A DCG with maintained and exchanged-partner contact changes removed comprises disconnected two-node subgraphs. Each subgraph has a single AB edge for conformation 1 or a single BA edge for conformation 2 and these are paired off to count the number of exchanged-pair contact-changes. Let n_1 be the number of remaining conformation 1 contacts after the maintained and the exchanged-partner contact-changes have been removed, and likewise n_2 be the number of remaining conformation 2 contacts. The number of exchanged-pair contact-changes was taken to be $N_{exchpair} = min(n_1, n_2)$. In a DCG with maintained, exchanged partner and exchanged-pair contact-changes removed there are only two-node subgraphs of one type left, either AB or BA. These represent the new contact-changes. The number of new contact changes, N_{new}, is then given by $N_{new} = n_1 - N_{exchpair}$ or

$N_{new} = n_2 - N_{exchpair}$, the former if $n_1 \geq n_2$, the latter if $n_2 \geq n_1$. In the case of (Figure 5.8) this DCG would be characterized as having no maintained, exchanged-pair and new but 2 exchanged-partners. In the case of Autolysin (Figure 5.6) there are no exchange-pairs but 1 each: exchange-partner, new and maintained and in the more elaborate case of DNA Topoisomerase III (Figure 5.5) there are 7 maintained, 13 exchanged-partner, 7 exchanged-pair and 4 new contact changes.

5.2.3 Domain Movement Classification

Domain movements can be classified according to whether the contact-change categories are empty or not. For the total number of contact-change types across all proteins in the dataset,

there are 6809 new, 6077 maintained, 1446 exchanged-pair and 1149 exchanged-partner.

$$\begin{aligned}
&\{N_{maint} = 0, N_{exchpart} = 0, N_{exchpair} = 0, N_{new} = 0\} = \text{Pure No Contacts} \\
&\{N_{maint} \geq 1, N_{exchpart} = 0, N_{exchpair} = 0, N_{new} = 0\} = \text{Pure Maintained} \\
&\{N_{maint} = 0, N_{exchpart} \geq 1, N_{exchpair} = 0, N_{new} = 0\} = \text{Pure Exchanged-Partner} \\
&\{N_{maint} = 0, N_{exchpart} = 0, N_{exchpair} \geq 1, N_{new} = 0\} = \text{Pure Exchanged-Pair} \\
&\{N_{maint} = 0, N_{exchpart} = 0, N_{exchpair} = 0, N_{new} \geq 1\} = \text{Pure New} \\
&\{N_{maint} \geq 1, N_{exchpart} \geq 1, N_{exchpair} = 0, N_{new} = 0\} = \text{Combined Maintained \& Exchanged-Partner} \\
&\{N_{maint} \geq 1, N_{exchpart} = 0, N_{exchpair} \geq 1, N_{new} = 0\} = \text{Combined Maintained \& Exchanged-Pair} \\
&\{N_{maint} \geq 1, N_{exchpart} = 0, N_{exchpair} = 0, N_{new} \geq 1\} = \text{Combined Maintained \& New} \\
&\{N_{maint} = 0, N_{exchpart} \geq 1, N_{exchpair} \geq 1, N_{new} = 0\} = \text{Combined Exchanged-Partner \& Exchanged-Pair} \\
&\{N_{maint} = 0, N_{exchpart} \geq 1, N_{exchpair} = 0, N_{new} \geq 1\} = \text{Combined Exchanged-Partner \& New} \\
&\{N_{maint} = 0, N_{exchpart} = 0, N_{exchpair} \geq 1, N_{new} \geq 1\} = \text{Combined Exchanged-Pair \& New} \\
&\{N_{maint} \geq 1, N_{exchpart} \geq 1, N_{exchpair} \geq 1, N_{new} = 0\} = \text{Combined Maintained, Exchanged-Partner} \\
&\text{\& Exchanged-Pair} \\
&\{N_{maint} \geq 1, N_{exchpart} \geq 1, N_{exchpair} = 0, N_{new} \geq 1\} = \text{Combined Maintained, Exchanged-Partner} \\
&\text{\& New} \\
&\{N_{maint} \geq 1, N_{exchpart} = 0, N_{exchpair} \geq 1, N_{new} \geq 1\} = \text{Combined Maintained, Exchanged-Pair} \\
&\text{\& New} \\
&\{N_{maint} = 0, N_{exchpart} \geq 1, N_{exchpair} \geq 1, N_{new} \geq 1\} = \text{Combined Exchanged-Partner, Exchanged-Pair} \\
&\text{\& New} \\
&\{N_{maint} \geq 1, N_{exchpart} \geq 1, N_{exchpair} \geq 1, N_{new} \geq 1\} = \text{Combined All}
\end{aligned} \tag{5.2}$$

The binary option of each change category being empty or not empty means there are 2^4, so 16, classes (Equation 5.2). These can be further subdivided into a "pure" class, a "dual hybrid" class, a "triple hybrid" class and a "total combined" class. The pure class includes the "no contacts" category where there are no contacts between these domains in either conformation. The total combined class is for protein cases where all four contact-change types occur, making it difficult to categorize into a particular motion. There are 4 pure class categories, 6 dual hybrid categories and 4 triple hybrid categories (where there is at least one empty and one non-empty contact-change group). (Table **??**).

Class	Number of Examples	Number of Shear (% out of 20)	Number of Hinge (% out of 43)
Pure No Contacts	413	-	-
Pure Maintained	56	0 (0%)	0 (0%)
Pure Exchanged-Partner	3	0 (0%)	0 (0%)
Pure Exchanged-Pair	9	0 (0%)	0 (0%)
Pure New	376	4 (20%)	15 (35%)
Combined Maintained & Exchanged-Partner	10	1 (5%)	0 (0%)
Combined Maintained & Exchanged-Pair	44	0 (0%)	1 (2%)
Combined Maintained & New	225	1 (5%)	4 (5%)
Combined Exchanged-Partner & Exchanged-Pair	1	0 (0%)	0 (0%)
Combined Exchanged-Partner & New	34	1 (5%)	0 (0%)
Combined Exchanged-Pair & New	78	1 (5%)	6 (14%)
Combined Maintained, Exchanged-Partner & Exchanged-Pair	35	2 (10%)	1 (2%)
Combined Maintained, Exchanged-Partner & New	126	3 (15%)	5 (12%)
Combined Maintained, Exchanged-Pair & New	137	3 (15%)	2 (5%)
Combined Exchanged-Partner, Exchanged-Pair & New	53	0 (0%)	3 (7%)
Combined All	222	4 (20%)	6 (14%)
Totall	1822	20 (100%)	43 (100%)

Table 5.1 Sixteen classes with total numbers of proteins and the percentage of Shear or Hinge found in the DBMM.

Chapter 6

Results: Predicting Hinge and Shear

6.1 Translation or Rotation in domain movements

The original Logistic Regression analysis proves to be a good predictor of hinge and shear in protein domain movements. Whether intended or not, the term ‘shear’ is often interpreted to mean a relative translation of the domains whereas the term “hinge” is a pure rotation. Our predictor of hinge and shear allows us to create a large data set of predicted hinge and shear movements in proteins, on which it is possible to test for variance between the two sets in this regard.

6.1.1 Dynamic Contact Graph Analysis

The new method of DCG’s has provided a new way of investigating domain movements in proteins. When the DCG is broken down into the number of elemental contact changes this data could be used as input for an alternative logistic regression analysis to the one described above.

6.1.2 Regression Analysis

Unlike the original logistic regression analysis with shear and hinge which used just a two component vector (Equation 3.21) a four component vector is used, which incorporates all four elemental contact changes where N_i represents a four-component vector with $N^i_1 = N^i_{maintained}, N^i_2 = N^i_{exchange-partner}, N^i_3 = N^i_{exchange-pair}$ and $N^i_4 = N^i_{new}$

where $N^i_{maintained}, N^i_{exchange-partner}, N^i_{exchange-pair}$ and N^i_{new}

signify $N_{maintained}, N_{exchange-partner}, N_{exchange-pair}$ and N_{new} in domain movement i. Let $t^i = 0$ when the DBMM assignment for domain movement i is predominantly hinge, and $t^i = 1$ when the DBMM assignment for domain movement i is predominantly shear. Given labeled training data (see appendix1 A) $D = \left\{ \left(\mathbf{N^i}, t^i \right) \right\}$ regularised logistic regression constructs a decision rule that can be used to distinguish between objects belonging to two classes (Equation 3.21). This is the same procedure as was employed for the first regression analysis with shear and hinge, however in this case **w** is now a four-component vector of regression coefficients, and b is an unregularised scalar bias parameter, with the optimal value of the regression coefficients still determined by minimising the regularised cross-entropy training criterion as before (Equation 3.22).

6.1.3 Hinge & Shear analysis

Prior to any further analysis 412 cases in the NRDB2d where N = 0, were omitted, i.e. those cases where $N_{maintained}, N_{exchange-partner}, N_{exchange-pair}$, and N_{new} are all equal to zero. These are cases where there are no contacts between the two domains in both conformations, and are neither hinge nor shear (indeed there are no "non-contacting" cases present in the 77 DBMM examples). As was the case in the original ROC analysis of the DBMM and the NRDB2d. 37 "predominantly shear" domain movements in the DBMM were shared with 21 in NRDB2d, and of the 75 "predominantly hinge" domain movements in the DBMM, 41 were also in NRDB2d. The DynDom program was also implemented on PDB structures not

found in the NRDB2d but were found in the DDMM, giving an extra 2 example to add to the 21 shear cases and an additional 13 to the 41 in the hinge cases. Logistic regression produced the following model:

$$y(N) = \frac{1}{1+e^{\alpha}} \tag{6.1}$$

$$\alpha = -0.2387N_{Maintained} - 0.0356N_{exchange-partner} + 0.4249N_{exchange-pair} + 0.2122N_{New} + 0.1467 \tag{6.2}$$

In order to determine whether this model corresponds well to the DBMM assignments a ROC curve was determined (Figure 6.1A). The area under the ROC curve is 0.83 which like the original hinge and shear ROC analysis specifies that the logistic function is a good interpreter of hinge and shear movements. A leave-one-out cross-validation approach was also used (Figure 6.1B). The area under this ROC curve is 0.77 which also confirms that the logistic function is a good predictor of hinge and shear. When comapred to the two compoent analysis done previously this gave an almost identical result.

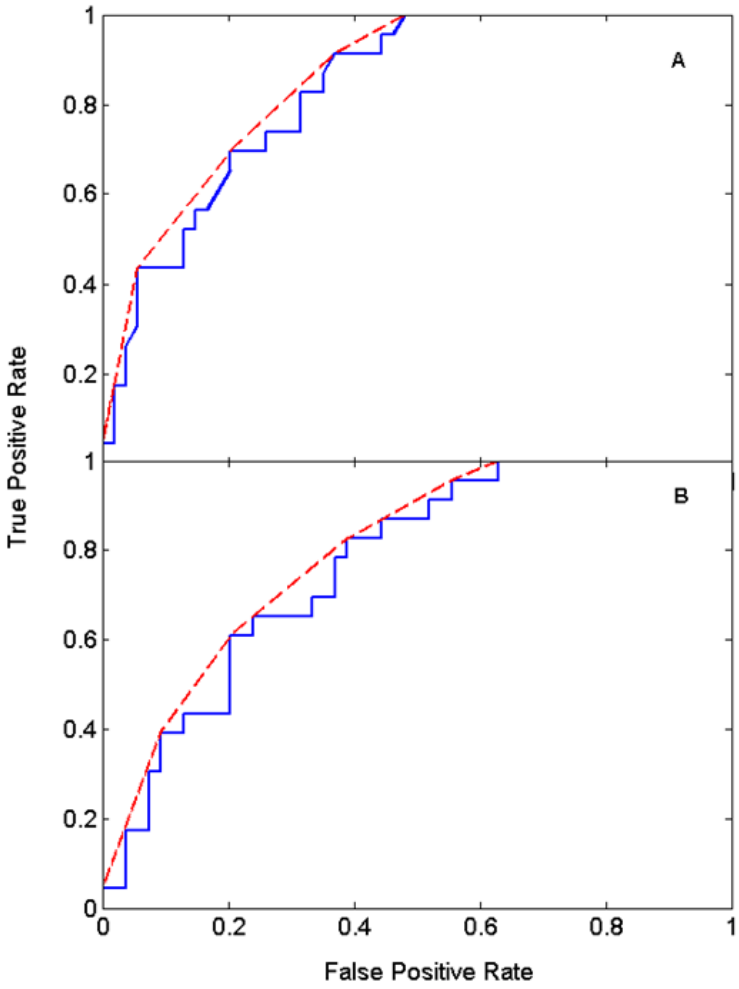

Fig. 6.1 ROC curves for the prediction of hinge and shear [A] Regular ROC // [B] Leave-one-out cross-validation ROC.

The logistic regression model was applied to 1410 movements (excluding the non-contacting cases) in the NRDB2d. (Figure 6.2A) shows a histogram for the prediction values y. No clustering is observed but peaks are present at certain values of y. The peaks labeled a, b, c, d, e present different instances of "pure new" contacts where $N = (0,0,0,N_{new})$ with $N_{new} = 1,2,3,4,5$ respectively. The pure new class is the second largest class after "non contact" signifying one of the conformations makes no contact with the other domain at all but the other conformation does, indicating a hinge movement.

Domain movements were put into three divisions: "Hinge", defined as $0 \leq y \leq 0.45$ "Shear", defined as $0.55 \leq y \leq 1.0$ and "Mixed", given by $0.45 < y < 0.55$. For the sake of comparing the properties of the two main classes, Hinge and Shear, it is desirable to have a high precision, calculated as the proportion of cases predicted correctly to be in that category (true positives) to the total number cases predicted to be in that category. Of the 61 DBMM cases predicted Hinge, 48 were "predominantly hinge" according to DBMM resulting in a precision of 79%. The precision calculation of predicted Shear only gives 9 DBMM cases, but 12 of them were categorized as predominantly shear by DBMM giving a precision of 75%. Unlike the previous logistic regression analysis on just shear and hinge the natural boundary of 0.5 (Hinge given as $0 \leq y \leq 0.5$and Shear $0.5 < y \leq 1.0$) is not favored with the precision for the Shear class dropping below 70% providing poor statistical results. Therefore the 0.45 and 0.55 divisions as classification limits, highlight it is possible to allocate hinge and shear to domain movements automatically with a high degree of success and correspondence with classifications assigned using the intuitive method. This predictor method when applied to the 1410 cases, produced the following results: 884 Hinge (63%), 361 Shear (26%) and 165 Mixed (12%). When considering all 1822 domain movements, which include the non-contact set with 23%, 49% are Hinge, 20% are Shear, and 9% Mixed. This results in a tenfold growth in the number of cases previously available, permitting statistical methods to be used to study hinge and shear mechanisms.

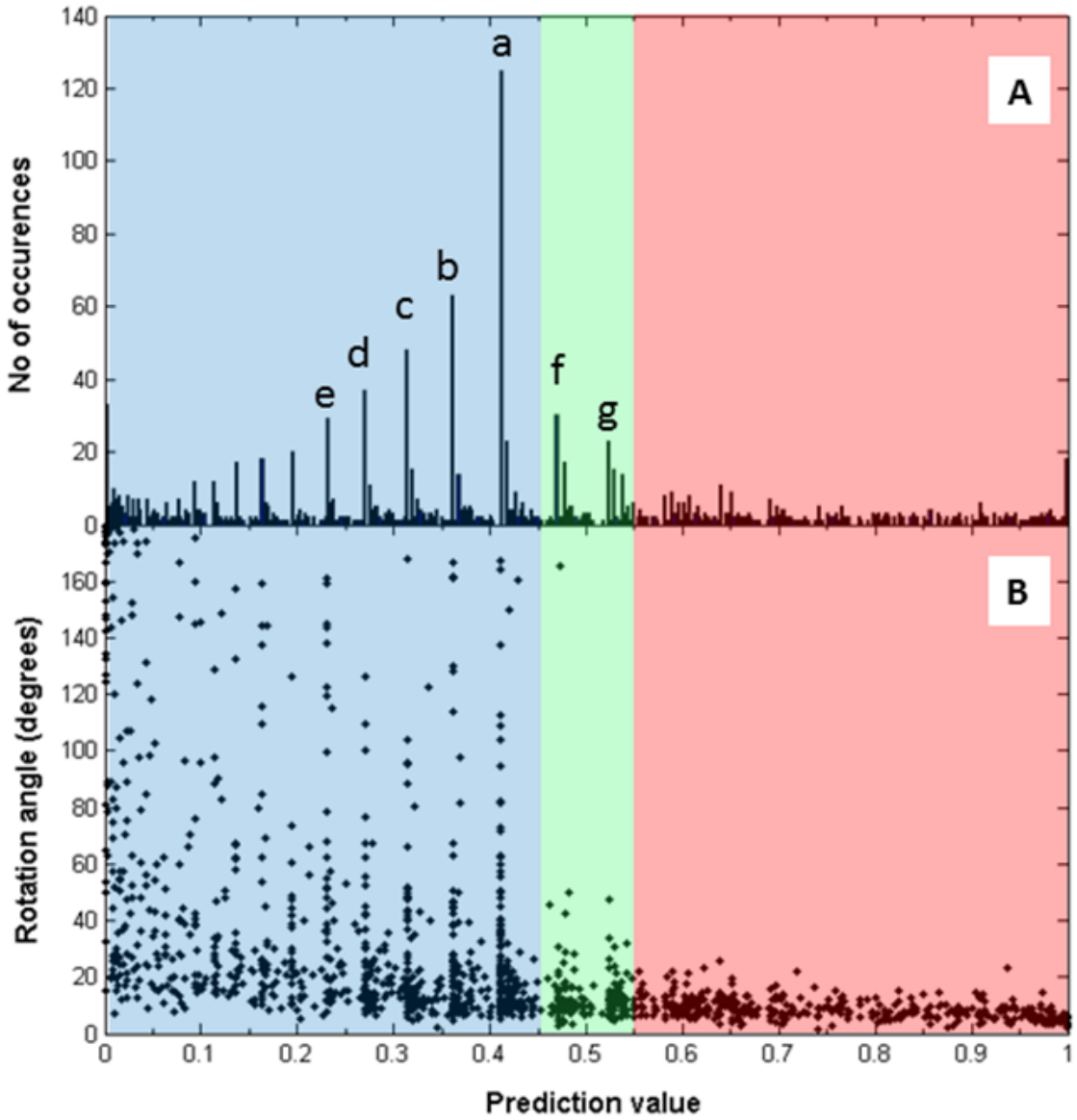

Fig. 6.2 Prediction value distributions. Blue region = "Hinge", green = "Mixed" and red = "Shear" (A) Histogram of prediction values. The spikes indicated by ["a" = N = (0 0 0 1)], ["b" = N = (0 0 0 2)], ["c" = N = (0 0 0 3)], ["d" = N = (0 0 0 4)], ["e" = N = (0 0 0 5)], ["f" = N = (1 0 0 1)] and ["g" = N = (1 0 0 0)] (B) The rotation angle vs. prediction value plot. The same peaks can be seen and give an explanation for their presence. The peak at "a" for prediction value 0.411 relates to N = (0 0 0 1), and signifies a great number of domain movements with different angles of rotation that are all capable of breaking a single residue contact pair.

For these cases the larger N_{new} the more "hinge" they seem to become in terms of their y value (decreasing with increasing N_{new}), for all of these, $y < 0.45$. The peak f, at y = 0.470, is

somewhat of an anomaly because of the prevalence of instances with N = (1 0 0 1) from the hybrid "Combined maintained new" category (the third biggest). Moving into the "mixed" division, peak g, y = 0.523, is from "Pure maintained" when N = (1 0 0 0) with a single pairwise residue contact maintained in the domain movement. The histogram (Figure 6.2B) highlights the rotation angle plotted against the prediction value, which can be compared to the initial Nvalue vs. rotation angle in the original residue contact analysis (Figure 4.7). The colour coded divisions show the different movements expected with the blue region: Hinge, green: Mixed and red: Shear. The association between angle of rotation and y value is also similar: the line shows that as the angle of rotation increases the prediction value decreases, meaning the motions become more hinge. The blue hinge division also highlights it is the large rotations which take place under 0.4, with peaks from (Figure 6.2A) belonging to "Pure New", this means 80% of the peaks is when N_{new} is greater than $N_{maintained}, N_{exchange-partner}$ and $N_{exchange-pair}$.

6.1.4 Rotation angle in Hinge and Shear movements

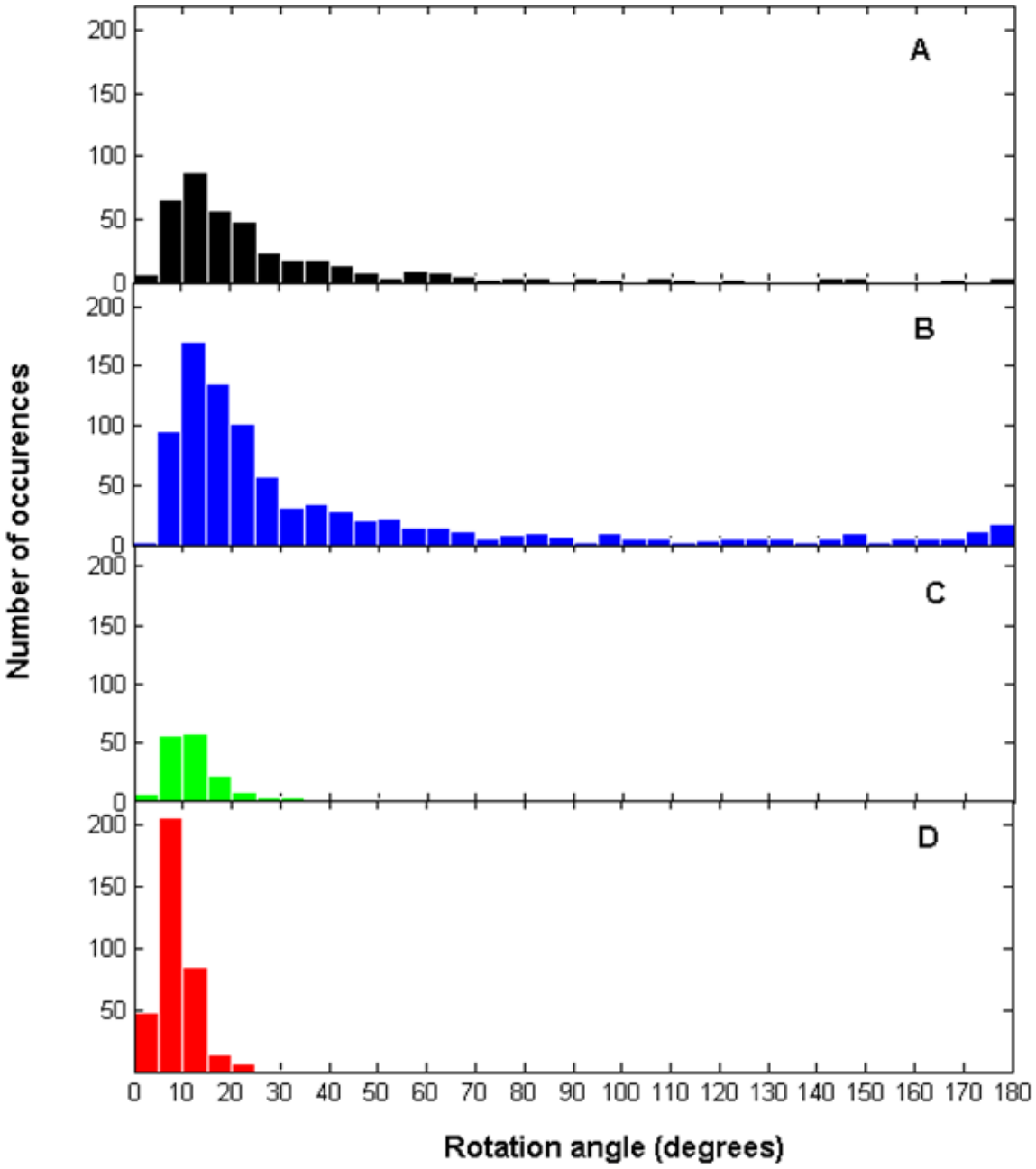

Fig. 6.3 Histograms of rotation angles. (A) No contact, (B) Hinge, (C) Mixed, (D) Shear.

(Figure 6.3) shows that Shear rotations do not go above 25 degrees. There is a predominance in the number of maintained, the number of exchanged-partner cases, signifying that a preserved-interface movement would be restricted to 25 degrees. It is also apparent in these

histograms that there is a small rise in Hinge cases where rotation angle approaches 180 degrees. Some can be accredited to "domain swapping" [9], a mechanism in oligomeric assembly formation. Domain swapping is a mechanism for forming oligomeric assemblies. In domain swapping, a secondary or tertiary element of a monomeric protein is replaced by the same element of another protein. Domain swapping can range from secondary structure elements to whole structural domains. It also represents a model of evolution for functional adaptation by oligomerisation, e.g. oligomeric enzymes that have their active site at subunit interfaces. Figures 6.1 and 6.3 indicate that rotation angle can be used as a measure of predicting if a domain movement is Hinge or Shear (Figure 6.4). The continuous blue line gives the percentage of cases for all three sets (Hinge, Shear and Mixed) with rotation angles larger than or equivalent to the rotation angle at a point on the line, showing that (discounting non-contact cases) the set of domain movements with a rotation angle of at least 10 degrees, 80% are Hinge. The broken red line provides the proportion of cases for all three sets (Hinge, Shear and Mixed) with rotation angles less than or equivalent to the rotation angle at that point on the line. Highlighting those domain movements (discounting non-contact cases) with a maximum rotation angle of 6 degrees, 80% are Shear.

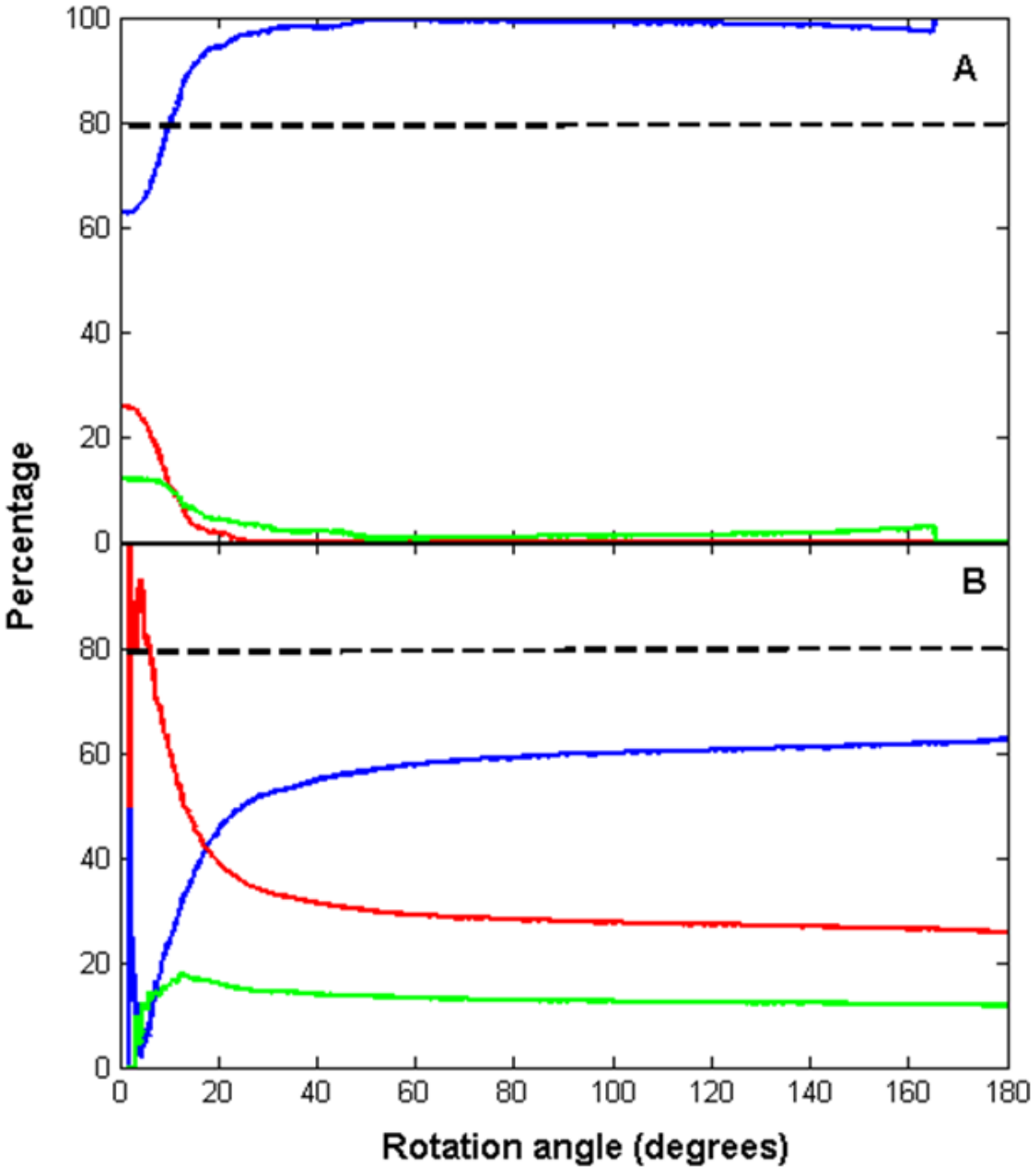

Fig. 6.4 Predictive value of angle of rotation. Blue lines = "Hinge", green lines = "Mixed" and red lines = "Shear". (A) Any given point on a line gives proportion $(\%)$ of domain movements (omits non-contact cases) with rotation angles $\geq$ to that given at the point, that are from the movement specified by the colour of the line from (Figure 6.3). (B) Any given point on a line gives the proportion $(\%)$ of domain movements (omits noncontact cases) with rotation angles $<$ that given at the point, that are from the movement specified by the colour of the line from (Figure 6.3)

6.1.5 Translation in domain movements

DynDom gives the rotation angle and also the translational displacement along the axis that occurs in the screw movement. If the movement is a pure rotation about an axis then this screw axis is the rotation axis. If a body undergoes a rotation about a hinge but also undergoes a translation in the plane of the rotation, then the interdomain screw axis will not coincide with the hinge. Thus we test for the screw axis being located outside the body of the protein. If this is the case then we can be sure that there is no control over the rotation being exercised at the axis location and consequently the rotation must be accompanied by a translation in the rotation plane. Of the 361 Shear cases, only 5 (1.4%) have the screw axis outside the body of the protein (a cut-off distance of 5.5 Å between the axis and any atom of the protein was used). Of the 884 Hinge examples, only 9 (1.0%) had the axis outside the body of the protein. These percentages show translational movements to be extremely infrequent. This suggests that the concept of a shear movement is a false analogy. Below we show that statistically there is no significant different between hinge and shear in this regard.

6.1.6 Significance Testing

Our data are divided into two main sets, Shear and Hinge, between which we are testing whether there is a significant difference in the values associated with a particular feature. Given our sets are large a Normal approximation to the Binomial is made. Let $N_H, \bar{X}_H$ and σ_H denote the number examples, the mean value of a particular feature and its standard deviation, respectively, in the Hinge set and $N_S, \bar{X}_S$ and σ_S the equivalent quantities for the Shear set. The z-value is calculated as:

$$z = \frac{\bar{x}_S - \bar{x}_{HS}}{\sqrt{\frac{\sigma_S^2}{N_S} + \frac{\sigma_H^2}{N_H}}} \tag{6.3}$$

In the case where we count the number of examples in a set possessing a particular feature

of interest we use the following test. Let n_H be the number of examples in the Hinge set that possess the feature and n_S be the number in the Shear set that possess the feature. Under the null hypothesis of there being no difference between the two sets with respect to this feature, the probability of its occurrence is:

$$p = \frac{n_H + n_S}{N_H + N_S} \tag{6.4}$$

and the z-value for the difference in the proportions amongst the two sets for this feature would be given by the following equation where $q = 1 - p$:

$$z = \frac{\frac{n_S}{N_S} - \frac{n_H}{N_H}}{\sqrt{pq\left(\frac{1}{N_H} + \frac{1}{N_S}\right)}} \tag{6.5}$$

Using the test of (Equation 6.5) a z-value of 0.56 was found giving $p(z \geq 0.56)$ = 29% for the probability that this difference $(1.4\%$ vs. $1\%)$ or larger happens by chance, meaning it is just as plausible for shear motions to have the axis within the body of the protein as hinge movements, indicating shear movements cannot be regarded as having one domain translate relative to the other. However, there could still be a rotation about an axis within the body of the protein which is accompanied by a large translation along the axis direction. Considering translation in the axis direction, the mean absolute value for the Hinge data is 1.47Å $\left(\text{standard deviation} = 3.1\text{Å}\right)$ whereas for the Shear data the mean is 0.35Å $\left(\text{standard deviation} = 0.37\text{Å}\right)$. Meaning there is more translation along the axis in Hinge than Shear, but this could be due to the fact that the rotations are greater amongst the Hinge set. Examining the pitch would make it more sense. The mean absolute value of the pitch for Hinge is $0.043\text{Å}/\text{degree}$ $\left(\text{standard deviation} = 0.095\text{Å}/\text{degree}\right)$ while for Shear the mean is $0.044\text{Å}/\text{degree}$ $\left(\text{standard deviation} = 0.058\text{Å}/\text{degree}\right)$. Using the test of (Equation 6.3) it was found that this difference was not significant $(p = 58\%)$. The presence of an "effective hinge axis" is another way of comparing shear and hinge. For Shear, 61 cases lacked an

effective hinge axis (16.8%) while the equivalent assessment for Hinge gave 117 (13.2%) with p = 4.7%. At the 5% level this would be significant, so advocates in the case of Shear, contacts at the preserved domain boundary help control the domain movement, whereas in Hinge it is more likely to be the backbone connections between the domains.

6.1.7 Twisting movement analysis

The concept of a shear motion according to Gerstein et al. is consistent with a sliding movement. In fact according to the assumptions given above concerning idealised domains and their movements, an exchanged-partner contact change is compatible with a "sliding twist" movement. The elemental contacts changes new or exchanged-pair would only ensue when two domains make an open-closed or see-saw domain movement under the same assumptions. Thus one would expect twisting movements to be more prevalent in the Shear cases rather than the Hinge. In the Shear set, 114 have 'predominantly twisting movement' according to DynDom at 32.0% while the equivalent assessment for Hinge is 192 at 21.7%. With p = 0.012% this variation is decidedly significant, demonstrating that twisting movements are more predominant in the Shear set.

6.2 DCG and Hinge Shear web content

6.2.1 DCG

A website was constructed which presented the new DCG data, organized according to the 16 class domain movement classification system (http://www.cmp.uea.ac.uk/dyndom/class16). Each protein is displayed with its protein name, the two corresponding PDB ID codes along with their chain identifiers and a link to its individual webpage (Figure 6.5) with a bar chart which identifies the number of elemental contacts present, the DCG itself and Jmol animation, (http://jmol.sourceforge.net/) which loops between the two conformations highlighting the amino acids making contact between the domains (Figure 5.5). In addition the number of disconnected regions is given and links to the analogous DynDom pages, the "DynDom Movement Page" which presents further information on domain movement (giving details on sequence, domain locations, hinge axis position, residues in the hinge-bending regions, the angle of rotation, percentage closure) and a link to the protein's family, called the "DynDom Family Page" (which gives information on closely related structures and their domain movements) (Figure 5.5).

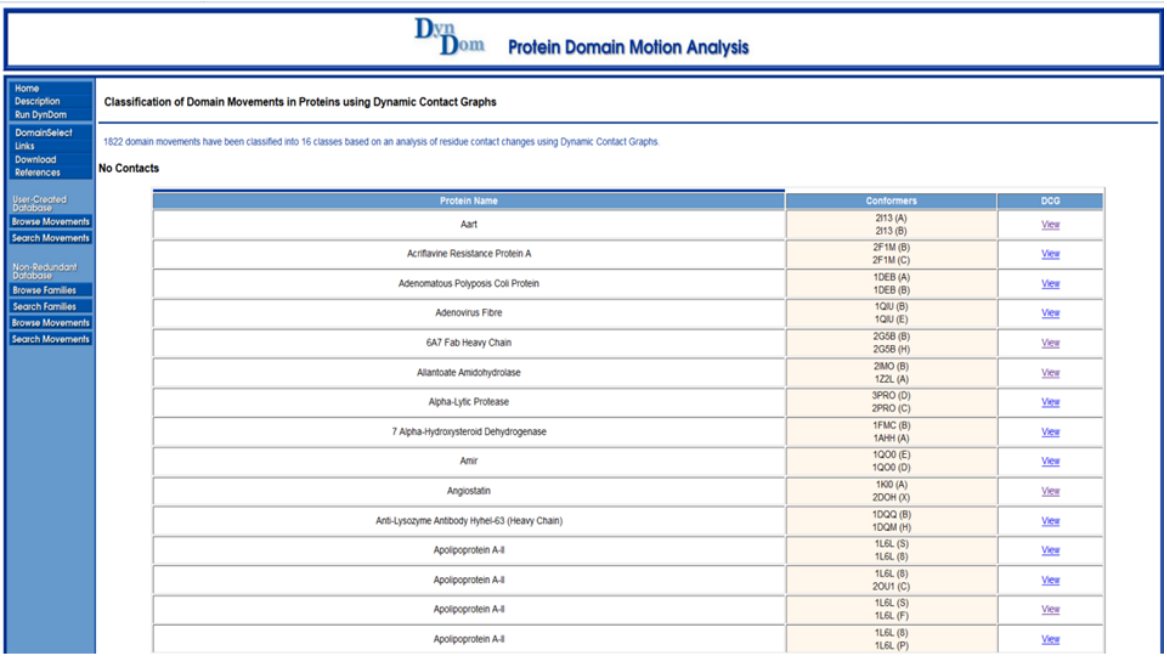

Fig. 6.5 Homepage for the DCG analysis across the NRDB2d.

6.2.2 Shear & Hinge

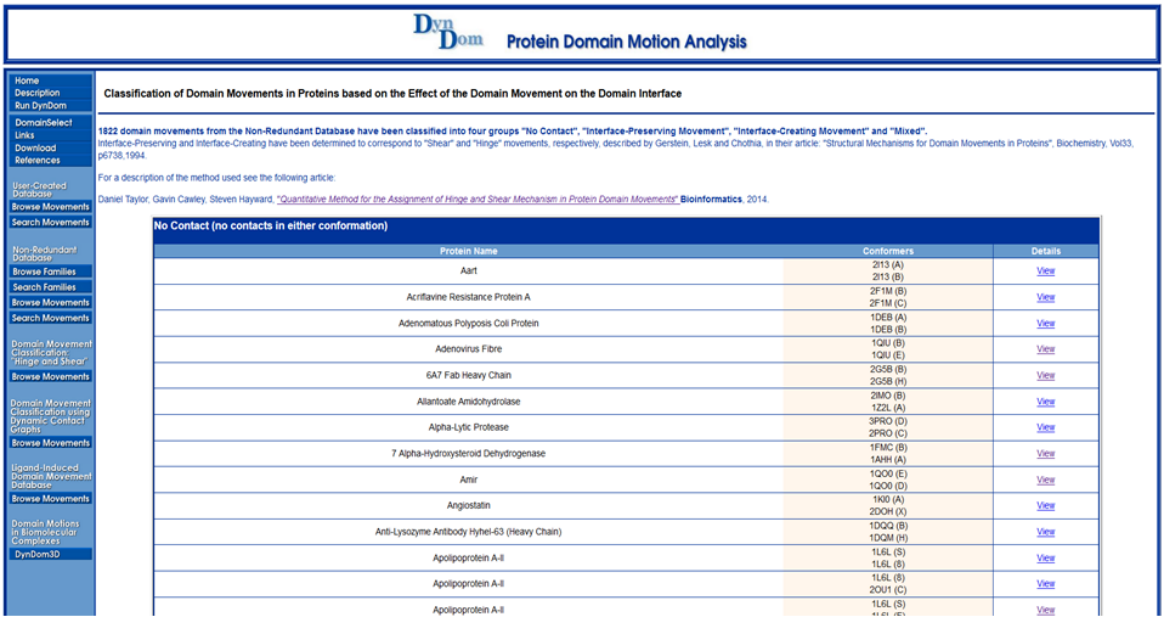

Fig. 6.6 Hinge and Shear Classification from DCG analysis homepage.

The results from the hinge and shear analysis can be found on a website very similar to the DCG classification (http://fizz.cmp.uea.ac.uk/dyndom/interface/) The categorisation of domain movements is simpler in this scheme, firstly there is No-contact, “Interface-preserving movement” (also known as Shear), “Interface-creating movement” (also known as Hinge) and Mixed (both Interface-preserving movement and Interface-creating movement) (Figure 6.7). As before, each entry within each subcategory presents the name of the protein, the PDB codes and chain identifiers and a link to its page. This page gives the name and PDB codes (with chain identifiers) of the protein, the angle of rotation (as calculated by DynDom), the prediction value, and the accompanying classification. The same Jmol animation of the two PDB conformations is used in these webpages as well. The links provided are also the same as the DCG web pages but with the addition of a link to the same DCG entry from the (http://www.cmp.uea.ac.uk/dyndom/class16) website called “Dynamic Contact Graph Page” for the corresponding protein (Figure 1.10 7).

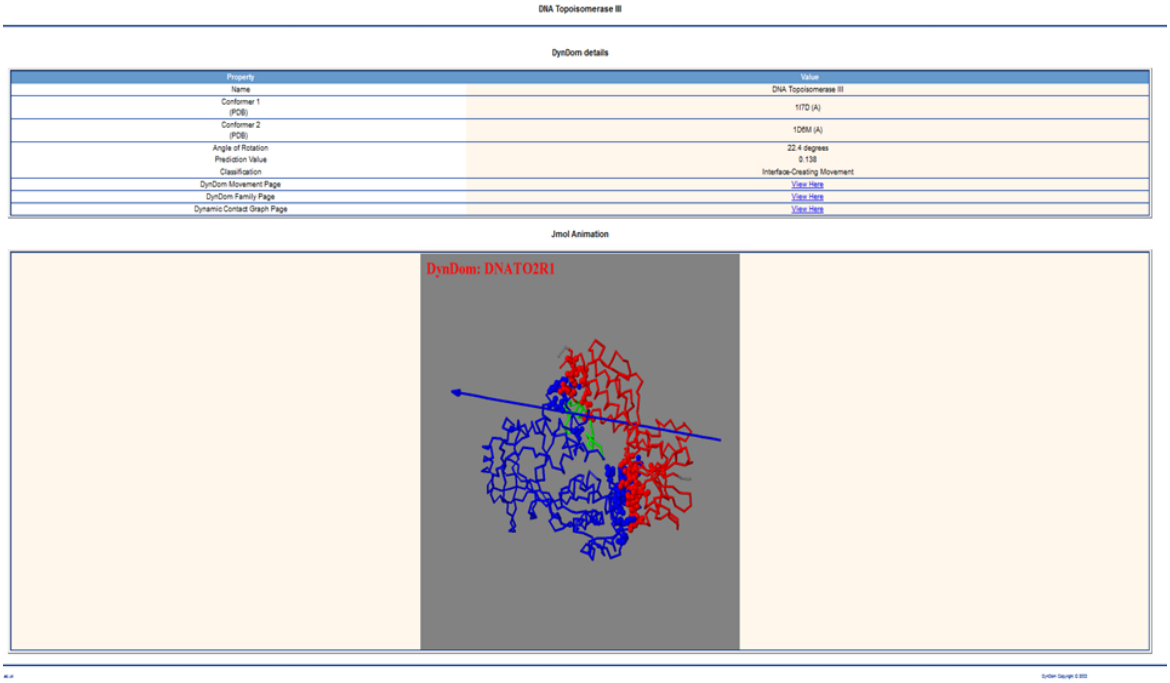

Fig. 6.7 Individual DCG analysis example with DNA Topoisomerase III in the "Interface-Creating Movement" subsection.

Chapter 7

Discussion and Conclusion

7.1 Dynamic Contact Graphs

A contact analysis has been used to categorise domain movements in proteins. This has been achieved with the identification of five types of elemental residue contact-changes. A true domain movement will be a complex set of different elemental contact-changes. The breakdown of a real domain movement into these elemental contact-changes is not straightforward. This, however, can be accomplished with the help of DCG's, which can be analysed to determine the number of different elemental contact-changes. This leads to a classification system comprising sixteen classes based on which elemental contact changes are present, or not. Each elemental contact-change type can be directly associated to a model domain movement. Each of the sixteen classes gives the degree to which each of the four elemental contacts makes to the overall movement. When only one is present, this is known as "pure" and is the most extreme case. In reality, a domain movement not in the pure class probably cannot be thought of as comprising a combination of the model domain movements associated with the corresponding elemental contact changes. The type of contact change made could be influenced by many different factors such as the size and flexibility of the residues, the position, amino acid content and local structure in the interdomain region, and

its closeness to the hinge axis. It has also proved useful to count the number of separated subgraphs in a DCG, as this gives the number of isolated regions (regions that have no contacts with other regions in both conformations) which may play a different role in the domain movement mechanism. In terms of energetics, which has been looked at before in terms of protein domain movements [82], the four elemental contact-changes could be used to describe the flow of energy when a protein domain movement is undertaken. In the case of the no contact class, no energy is required because no contact is made or broken. The "new" contact suggests energy needs to be inserted into or extracted from the system for the new contact be made or broken. The "maintained" contact, by its very nature, indicates a strong stabilization interface where either little or no energy change is needed. An "exchanged-partner" contact could suggest a low energy barrier because as a sliding motion takes place, as one contact breaks another is created. Finally the "exchanged-pair" contact could imply an energy barrier as when one interaction is broken before another is made, in a "see-saw" like motion. This idealised view of domain motions cannot do justice to the extremely intricate nature of a protein domain movement. One can look at domain movements from many other perspectives. For example, one can focus on how the type of protein/enzyme and how its functional purpose relates to the structural change [2] or how the proportion of hydrophobic surface area exposed changes as the protein moves between open and closed states [114].

The DCG method can be applied to any two domain protein with two structures solved in different conformations, with contact between the domains being made in at least one conformation. Another benefit of this new method is that it can be applied to individual examples of domain movements, which should provide a great deal of insight when examined by experts on the protein concerned. In essence the DCGs give a visual metaphor for the movement and its mechanism. Here we consider motifs that appear in DCGs indicating particular mechanisms:

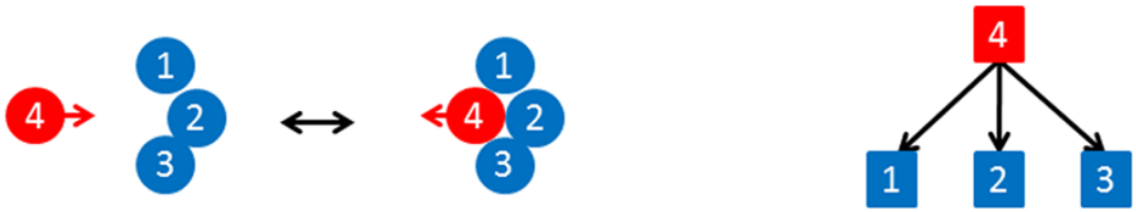

Fig. 7.1 Multiple New Domain Movement

Multiple new: A residue with no contact in one conformation moves into a pocket making multiple contacts in the other conformation. The associated graph is shown in (Figure 7.1) and is a clearly recognisable motif. The domain movement in aclacinomycin 10-hydroxylase provides an example (structural pair: 1XDS, chain A; 1QZZ, chain A).

Fig. 7.2 Linear Interlocking Movement

Linear Interlocking: A sequence of interlocking residues, depicted as a shear movement would have a graph, as shown in (Figure 7.2), with a series of doubly linked nodes. The doubly linked nodes give the visual metaphor of strong contacts between residues that cannot be broken. This motif is easily seen in a visual scan of a DCG. Tryptophanyl-tRNA synthetase (structural pair: 1MAU, chain A; 1I6M, chain A) provides an example.

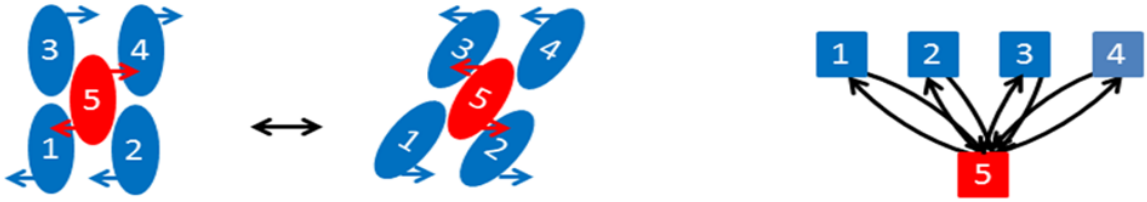

Fig. 7.3 Anchoring Residue Movement

Anchoring residue: A single residue maintains contact with a number of other residues during the domain movement, possibly acting as an anchor, as shown in (Figure 7.3). The

domain movement in glucokinase provides an example (structural pair: 1Q18, chain A; 1SZ2, chain B).

Fig. 7.4 Linear Slide Movement

Linear slide: A region from domain B (red in Figure 7.4) sliding on a region from domain A (blue) has a graph with a series of singly linked nodes with edges all pointing in the same direction. Consider the region from domain B sliding on the surface provided by the region of domain A with the direction of the edges indicating the direction of the movement of domain B going from conformation 1 to conformation 2, e.g. residue 4 is moving from residue 1 to residue 2. Again the graph gives a visual metaphor for a simple sliding movement and is an easily recognised motif. The domain movement in human IGG1 FC fragment provides an example (structural pair; 1E4K, chain B; 1IWG, chain A).

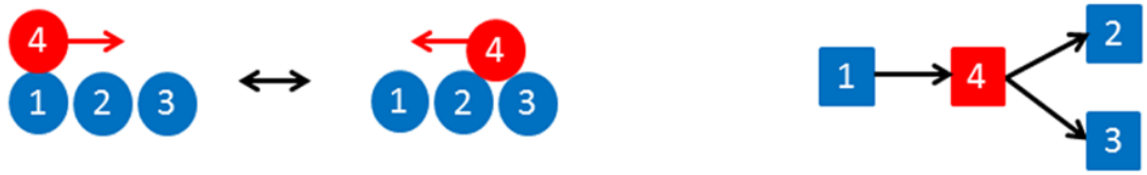

Fig. 7.5 Branched Slide Movement

Branched slide: If a residue in domain B makes a single contact with a residue in domain A, in conformation 1, but makes contact with two residues in domain A, in conformation 2, then the graph will have a branch, as shown in (Figure 7.5). The movement in a MHC class I molecule provides an example (structural pair: 1ZT7, chain C; 1MWA, chain I).

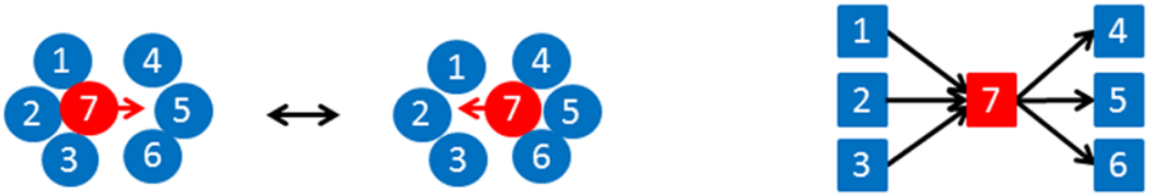

Fig. 7.6 Multiple-to-multiple Slide Movement

Multiple-to-multiple slide: If in conformation 1 a residue in domain B makes multiple contacts with residues in domain A, and moves to make multiple contacts with another region of domain A, in conformation 2, the graph will be as shown in (Figure 7.6). Again the graph provides a clear visual metaphor of the type of contact-change that occurs. NADH pyrophosphatase provides an example (structural pair: 1VK6, chain A; 2GB5, chain A).

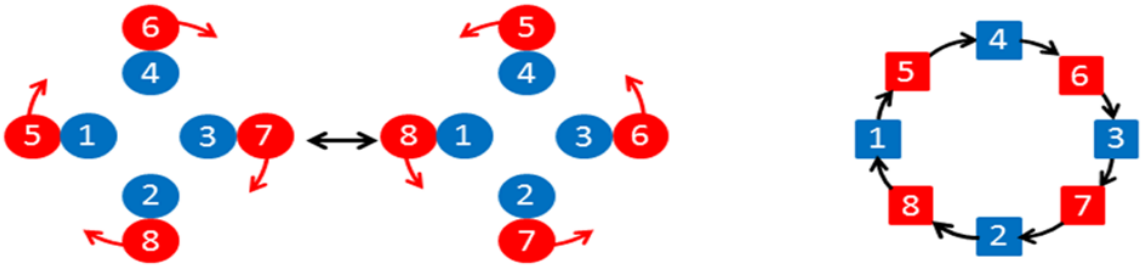

Fig. 7.7 Closed-cycle Slide Movement

Closed-cycle slide: If the two domains undergo a rotational motion, such that the two surfaces remain in contact, i.e. a twisting motion, and individual residues undergo a sliding movement where every residue makes a single contact in both conformations, then the graph will be a closed cycle, as shown in (Figure 7.7). The associated graph clearly indicates such a rotational motion, providing a visual metaphor for the movement and an easily recognisable motif. There are always even numbers of residues involved in this motif. The photosynthetic reaction centre from Thermochromatium tepidum provides an example (structural pair: 2EYT, chain A; 2EYS, chain A). As one might expect, the movement in this protein is predominantly a twist (with a 33.5% closure).

Fig. 7.8 Multiple See-saw Movement

Multiple see-saw: If a region makes contact in conformation 1 but not in conformation 2, and a completely separate region makes contact in conformation 2 but not in conformation 1, the graph will look like that shown in (Figure 7.8). This will occur when the domains undergo a see-saw motion. The associated graph provides a strong visual metaphor for a see-saw movement. The domain movement in maltodextrin binding protein provides an example (structural pair: 1MDP, chain 2; 2OBG, chain A).

The information used for the analysis comes directly from DynDom which analyses single subunits or monomeric proteins. Therefore the assumption being made with the DCG results, is that the contacts made between residues come from within the same subunit only as contacts with other subunits (if there are any) are not considered. Multimeric domain movements encompass both intrasubunit contact-changes and intersubunit contact-changes and consequently in the future intersubunit contact-changes must be incorporated. This can be done with the use of DynDom3D [95], which, with its new grid based method, can process domain movements in multimers. Perhaps the greatest potential benefit DCG analysis could have is when it is used alongside ligand binding data, as it is known that ligand binding often induces domain movement. Currently the DCG's do not include ligand-residue contact changes although it is not straightforward to include this. Further analysis of these DCG's and their corresponding structures could be used in Molecular Dynamics (MD) simulation, where principal component analysis provides eigenvectors from which two extreme structures can be generated, or Normal Mode Analysis (NMA) where a single normal mode eigenvector

can be represented by two structures from which residue contacts, or perhaps energy-based cut-offs, can be utilised to create the DCG.

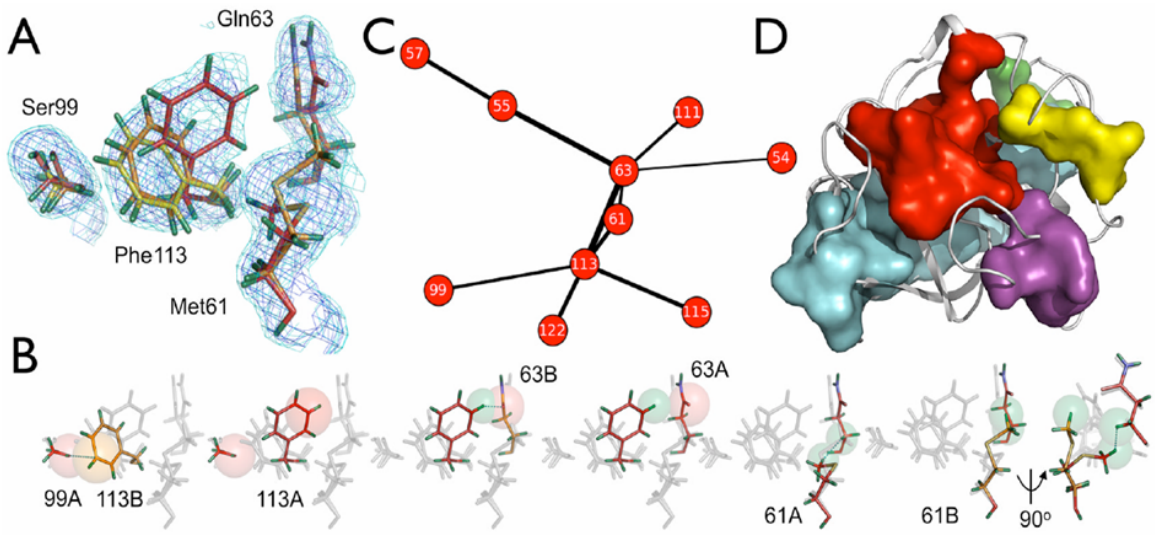

Fig. 7.9 Mechanisms for conformational exchange in Cyclophilin A (CYPA) (A) X-ray electron density map contoured at 1 σ (blue mesh) and 0.3 σ (cyan mesh) of CYPA is fit with discrete alternative conformations using qFit. Alternative conformations are coloured red, orange, or yellow, with hydrogen atoms added in green. (B) Pathway in CYPA: atoms involved in clashes are shown in spheres scaled to van der Waals radii and clashes between atoms highlighted by cyan dashes. (C) Networks identified by CONTACT are displayed as nodes connected by edges representing contacts that clash and are relieved by alternative conformations. The pathway in b forms part of the red contact network in CYPA and is highlighted by the dark purple edges. (D) The six contact networks comprising 29% of residues are mapped on the three dimensional structure of CYPA. The contact network shown in red overlaps with the dynamic network identified by NMR chemical shift perturbation and relaxation dispersion experiments [8].

Recent research has focused on identifying networks of conformational heterogeneous residues directly from high-resolution X-ray crystallography data using a new algorithm called CONTACT which automatically classifies residues that link functional sites, propagate chemical shift differences, and expose the structural mechanisms of mutations that affect rearrangements of the conformational ensemble [8]. These networks use electrodensity mapping comparisons to take into account the dynamic nature of the protein, and focus mainly on protein intramolecular fluctuations rather than domain movements. Therefore if

these networks were analysed alongside the DCG data, a great deal of information could be discovered regarding important residues through cross-referencing.

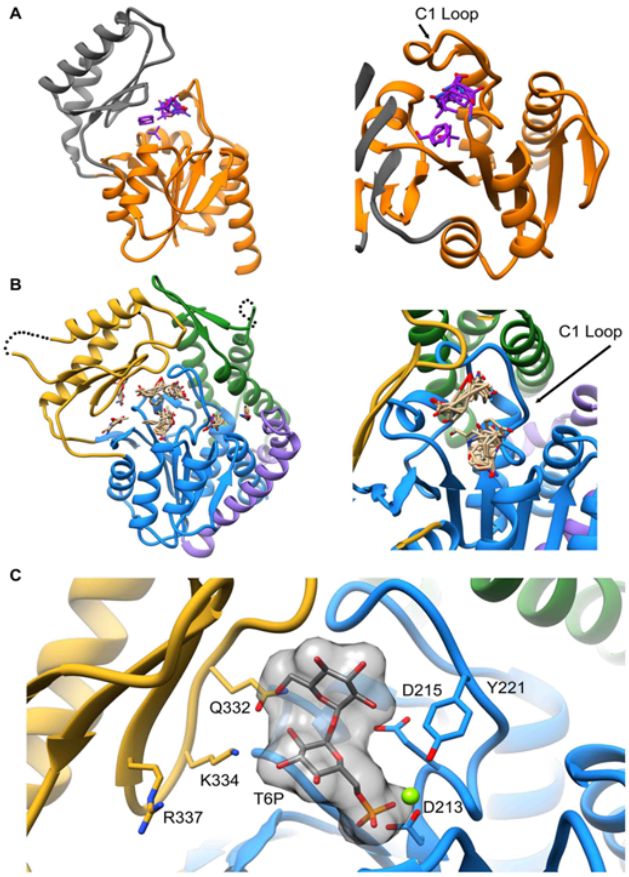

Fig. 7.10 The FTMap server was used to identify hot spots where protein-substrate interactions may occur. Analysis of the T6PP enzyme from T. acidophilium (1U02) (A), and B. malayi (B) reveal hot spots near the interface of the cap and core domains. These hot spots are cradled by the structurally conserved C1-Loop. T6P was placed manually into the active site of T6PP by coordinating the Mg2+ cation with the phosphate group (C). The residues identified as important via mutagenesis and kinetics are labeled and can be seen in proximity to the trehalose moiety [30].

The wide scope of DCGs has already been highlighted in the case of Trehalose-6-phosphate Phosphatase (T6PP), an enzyme used in the production of biological sugar. Trehalose is a natural alpha-linked disaccharide formed by an α,α-1,1-glucoside bond between two α-glucose units. When the crystal structure of the T6PP cap of *B. malayi* and that of *T. acidophilum* T6PP were superimposed and analysed using a DCG, it was found that the cap

rotates 45.6° in relation to the core. Placement of the cap in the predicted model positions the residues identified by mutagenesis and its DCG can highlight whether a specific residue is within contact distance of the predicted T6P model position [30].

DCGs can be usefully employed, not just in the study of proteins and their domain movements and can, be applied to any biomolecule where there are at least two conformations.

7.2 Hinge & Shear Analysis

The theory of hinge and shear mechanisms in protein domain movements was presented nearly twenty years ago [40]. The assignments of domain movements, according to these two criteria, has until this point been made on a subjective basis, requiring human curation. Not only is this prone to human error but has limited use as it can only be applied to a small number of cases. This method was developed before the recent boom in Bioinformatics technology and data. The PDB in the last 20 years has increased thirtyfold in terms of protein structural content and also in the number of domain movements. The NRDPDM database contains 2035 distinctive domain movements. It would be time-consuming undertaking to examine all of these domain movements using molecular graphics software to determine whether they have a hinge and/or shear mechanism. The qualitative process used until now requires instead a new quantitative method of analysis. There is a difficulty in taking a subjective method and creating from it an objective method that corresponds to it. To deal with this two aspects are discused: first an explanation of the objective method we used and secondly the classifications of the example data.

The numbers of occurrences of four different types of residue contact changes between domains were used as input for a logistic regression model to create a predictor for shear and hinge mechanism. The original hinge and shear assignments were used as training data for the logistic regression model. The results show that a new quantitative method for the assignment of hinge and shear mechanisms has been developed producing good results (as indicate by the high precision of predictions). This new technique has allowed classification of a much greater set of domain movements into Hinge and Shear, providing a tenfold increase in the number of cases formerly available. The larger set has allowed us to analyse a widely-held interpretation of the term “shear”, namely that domain closure takes place via translation of one domain relative to the other. Whilst accepting this is possible, the results

of this research have identified this is seldom the case, and no more likely to occur in the Shear Set than in the Hinge Set. In the light of this finding the term “shear movement” would be better renamed as “interface-preserving movement” and “hinge” as “interface-creating movement.” These terms are still compatible with the original model but are less likely to be misinterpreted.

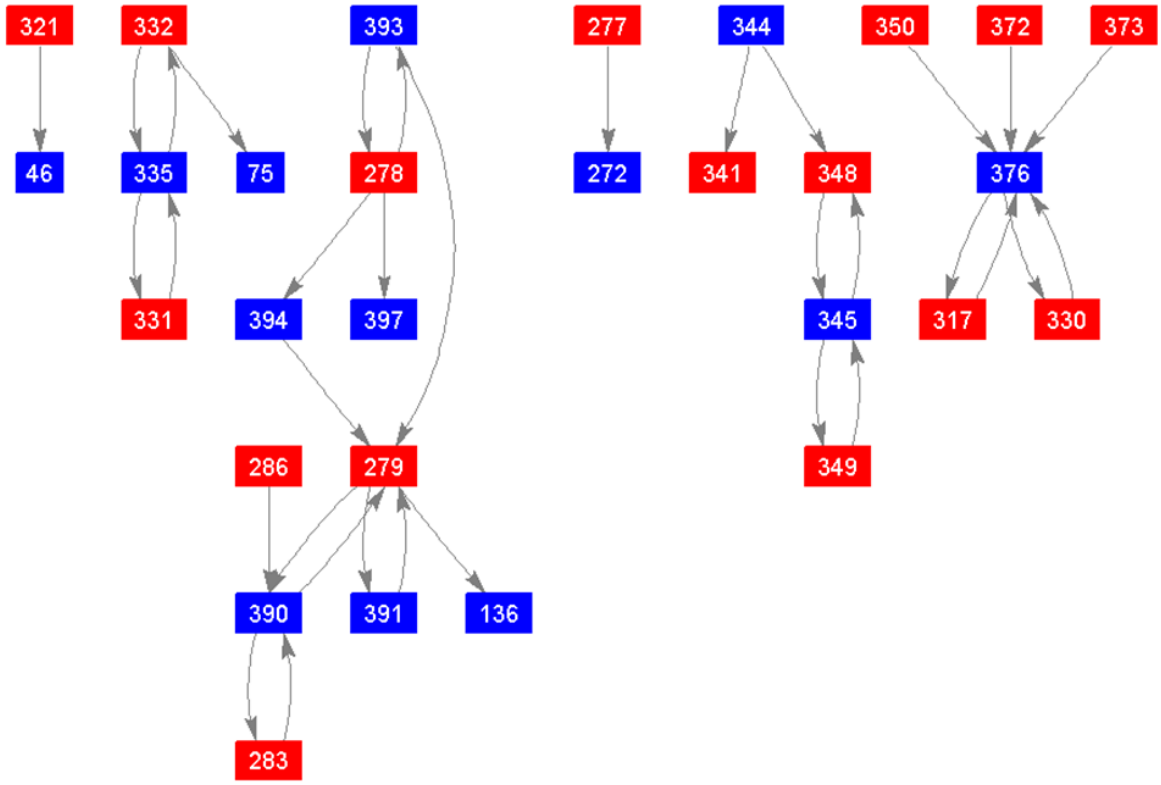

Fig. 7.11 The DCG for the domain movement between conformation 1 (PDB accession code: 1CTS) and conformation 2 (PDB accession code: 1CSH) in citrate synthase. A blue square corresponds to a residue in domain A and a red square corresponds to a residue in domain B with the residue number written in the square. An arrow from a residue in A to a residue in B indicates a contact between the residues in conformation 1. An arrow from a residue in B to a residue in A indicates a contact between the residues in conformation 2.

This research has identified a shear domain movement as being a movement which does not involve a substantial translation of the two domains but a rotation about an axis within the body of the protein, in much the same way as a protein undertakes a domain movement in a hinge motion. The logistic model shows that a high number of maintained and/or

exchanged-partner contact changes is an indicator for a shear movement, while a high number of exchanged-pair and new contact changes is a hinge movement indicator. In addition, the finding that there are more twisting movements in the Shear class than in the Hinge conforms to the idea that a twist movement is a rotation that can preserve an interface without relative translation. Nevertheless, not all predominantly interface-preserving movements occur via a twisting motion; several occur by a closure motion governed by rotation about precise hinges.

Citrate Synthase demonstrates that this "predominantly shear" movement, as assigned by the DBMM, is also an example of a protein that undertakes closure (84%) by a process of hinge bending, but preserves some part of the domain interface. Even though labeled as "predominantly shear" by DBMM it can still be defined as hinge-bending. It is in the "Mixed" class with a prediction value of 0.55, with marginally more interface preserving characteristics than interface creating. Figure 7.11 shows the DCG for citrate synthase. It has 10 maintained contact changes, 2 exchanged-partner contact changes, 2 exchanged-pair contact changes, and 6 new contact changes. It has a distinct hinge axis produced by mechanical hinges, one of which is a "hinged-loop" [48] a loop bordered by two bending regions through which the hinge axis goes through which helps to control the domain movement, similar to a hinge in a protein which undertakes closure via hinge bending e.g. in the case of Lactoferrin.

7.3 Conclusion

- This analysis led to the development of a novel directed graph concept termed the "Dynamic Contact Graph" (DCG) that represents how two protein domains move in relation to one another using residues contact changes. The method developed, has wide applicability and could even be applied outside of protein science.

- The DCG's were deconstructed by counting the number of instances of four types of elemental residue contact changes: new, maintained, exchanged partner and exchanged pair. This lead to a new classification scheme of 16 categories.

- These four types of residues contact changes were effectively combined by logistic regression using the training set of domain movements intuitively classified as hinge and shear at the Database for Molecular Movements (DBMM) to produce a predictor for hinge and shear. This predictor was applied to give a 10-fold increase in the number of examples over the number previously available with a high degree of precision.

- It is shown that overall a relative translation of domains is rare, and that there is no difference between hinge and shear mechanisms in this respect. However, the shear set contains significantly more examples of domains having a relative twisting movement than the hinge set.

References

[1] Abyzov, A., Bjornson, R., Felipe, M., and Gerstein, M. (2010). RigidFinder: A fast and sensitive method to detect rigid blocks in large macromolecular complexes. *Proteins*, 78:309–324.

[2] Amemiya, T., Koike, R., Kidera, A., and Ota, M. (2012). PSCDB: a database for protein structural change upon ligand binding. *Nucleic Acids Res*, D1:D554–D558.

[3] Andreeva, A., Holworth, D., Brenner, S., Hubbard, T., Chothia, C., and Murzin, A. (2004). SCOP database in 2004: Refinements integrate structure and sequence family data. *Nucleic Acids Research*, 32(90001):D226–D229.

[4] Andreeva, A., Howorth, D., Chandonia, J., Brenner, S., Hubbard, T., Chothia, C., and Murzin, A. (2007). Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Research*, pages 1–7.

[5] Aspvall, B., Plass, M., and Tarjan, R. (1979). A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123.

[6] Audle, D. and Smith, J. (2008). *http://people.virginia.edu/wrp/bioch503/bioch503_l7.html*.

[7] Barber, D. (2012). Bayesian reasoning and machine learning. *Cambridge University Press (2 Feb 2012)*.

[8] Bedem, H., Bhabha, G., Yang, K., Wright, P., and Fraser, J. (2013). Automated identification of functional dynamic networks from xray crystallography. *Nat Methods*, 10(9):896–902.

[9] Bennett, M., Choe, S., and Eisenberg, D. (1994). Domain swapping: Entangling alliances between proteins. *Proc. Nati. Acad. Sci. USA*, 91:3127–3131.

[10] Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., and Bourne, P. (2000). The protein data bank. *Nucleic Acids Res*, 28(1):235–242.

[11] Bishop, C. (2006). Pattern recognition and machine learning. *In the terminology of statistics, this model is known as logistic regression, although it should be emphasized that this is a model for classification rather than regression*, page 206.

[12] Branden, C. and Tooze, J. (1999). *Introduction to Protein Structure*. Garland Publishing Inc.

[13] Brünger, A. and Nilges, M. (1993). Computational challenges for macromolecular structure determination by x-ray crystallography and solution NMRspectroscopy. *Quarterly Reviews of Biophysics*, 26(1):49–125.

[14] Bru, C., Courcelle, E., Carrere, S., Beausse, Y., Dalmar, S., and Kahn, D. (2005). The ProDom database of protein domain families: more emphasis on 3d. *Nucleic Acids Research*, 33:D212–D215.

[15] Cantor, G. (1874). Ueber eine eigenschaft des inbegriffes aller reellen algebraischen zahlen. *J. Reine Angew. Math*, 77:258–262.

[16] Cauchy, A. (1813). Recherche sur les polyèdres - premier mémoire. *Journal de l'École Polytechnique*, 16:66–86.

[17] Cayley, A. (1857). On the theory of the analytical forms called trees. *Philosophical Magazine, Series IV*, 13(85):172–176.

[18] Cayley, A. (1875). Ueber die analytischen figuren, welche in der mathematik bäume genannt werden und ihre anwendung auf die theorie chemischer verbindungen. *Berichte der deutschen Chemischen Gesellschaft*, 8(2):1056–1059.

[19] Chasles, M. (1830). Notes on the general properties of a system of 2 identical bodies randomly located in space; and on the finite or infinitesimal motion of a free solid body. *Bulletin des Sciences Mathematiques, Astronomiques, Physiques et Chimiques*, 14:321–326.

[20] Concord.org (2013). http://www.concord.org/btinker/workbench_web/unitIV_revised/silk/silk_beta.html.

[21] Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2001). Introduction to algorithms. *Second Edition. MIT Press and McGraw-Hill*, pages 552–557.

[22] Cuff, A., Sillitoe, I., Lewis, T., Clegg, A., Rentzsch, R., Furnham, N., Pellegrini-Calace, M., Jones, D., Thornton, J., and Orengo, C. (2011). Extending CATH: increasing coverage of the protein structure universe and linking structure with function. *Nucleic Acids Research*, pages D420–D426.

[23] Debrunner, P. and Frauenfelder, H. (1982). Dynamics of proteins. *Annual Review of Physical Chemistry*, 33:283–299.

[24] Dengler, U., Siddiqui, A., and Barton, G. (2001). Protein structural domains: Analysis of the 3dee domains database. *Proteins: Structure, Function, and Bioinformatics*, 42(3):332–344.

[25] Dodd, L. and Pepe, M. (2003). Partial AUC estimation and regression. *Biometrics*, 59(3):614–623.

[26] Dodge, C., Schneider, R., and Sander, C. (1998). The HSSP database of protein structure-sequence alignments and family profiles. 26:313–315.

[27] Echols, N., Milburn, D., and Gerstein, M. (2003). MolMovDB: analysis and visualization of conformational change and structural flexibility. *Nucleic Acids Res*, 31(1):478–482.

[28] Even, S. (2011). Graph algorithms. *(2nd ed.), Cambridge University Press*, pages 46–48.

[29] Facey, G. (2009). *The Selective 1D Gradient NOESY http://u-of-o-nmr-facility.blogspot.co.uk/2009/05/selective-1d-gradient-noesy.html*.

[30] Farelli, J., Galvin, B.and Li, Z., Liu, C., Aono, M., Garland, M., Hallett, O., Causey, T., Ali-Reynolds, A., Saltzberg, D., Carlow, C., Dunaway-Mariano, D., and Allen, K. (2014). Structure of the trehalose-6-phosphate phosphatase from brugia malayi reveals key design principles for anthelmintic drugs. *PLoS Pathog*, 10(7):e1004245.

[31] Ferri, C., Hernandez-Orallo, J., and Salido, M. (2003). Volume under the ROC surface for multi-class problems. *Machine Learning: ECML*, pages 108–120.

[32] Fersht, A. (1999). *Structure and Mechanism in Protein Science - A Guide to Enzyme Catalysis and Protein Folding*.

[33] Finn, R., Marshall, M., and Bateman, A. (2004). iPfam: visualization of protein–protein interactions in PDB at domain and amino acid resolutions. *Bioinformatics*, 21(3):410–412.

[34] Finn, R., Mistry, J., Schuster-Böckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S., Sonnhammer, E., and Bateman, A. (2006). Pfam: clans, web tools and services. *Nucleic Acids Research*, 34(1):D247–D251.

[35] Finn, R., Mistry, J., Tate, J., Coggill, P., Heger, A., Pollington, J., Gavin, L., Gunasekaran, P., Ceric, G., Forslund, K., Holm, L., Sonnhammer, E., Eddy, S., and Bateman, A. (2010). The pfam protein families database. *Nucleic Acids Research*, 38:D211–D222.

[36] Flores, S. and Gerstein, M. (2007). FlexOracle: predicting flexible hinges by identification of stable domains. *BMC Bioinformatics*, 8(215):1–24.

[37] Garson, G. (2014). Logistic regression: Binary & multinomial (statistical associates "blue book" series book 2). *Statistical Associates Publishers; 2014 edition (29 Mar 2014)*.

[38] Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2013). Bayesian data analysis, third edition (chapman & hall/CRC texts in statistical science). *Chapman and Hall/CRC; 3 edition (5 Nov 2013)*.

[39] Gerstein, M. and Krebs, W. (1998). A database of macromolecular motions. *Nucleic Acids Res*, 26:4280–4090.

[40] Gerstein, M., Lesk, A., and Chothia, C. (1994). Structural mechanisms for domain movements in proteins. *Biochemistry*, 33:6739–6749.

[41] Goldstein, H. (1980). *Classical Mechanics*.

[42] Gondaliya, A. (2014). Regularization implementation in r : Bias and variance diagnosis. *http://pingax.com/regularization-implementation-r/*, May 22, 2014.

[43] Goto, S., Okuno, Y., Hattori, M., Nishioka, T., and Kanehisa, M. (2002). LIGAND: database of chemical compounds and reactions in biological pathways. *Nucleic Acids Res*, 30(1):402–404.

[44] Gould, R. (2012). Graph theory (dover books on mathematics). *Dover Publications Inc.; Reprint edition*.

[45] Guo, J., Xu, D., Kim, D., and Xu, Y. (2003). Improving the performance of DomainParser for structural domain partition using neural network. *Nucleic Acids Res*, 31(3):944–952.

[46] Hadley, C. and Jones, D. (1999). A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. *Structure*, 7(9).

[47] Harris, J., Hirst, J., and Mossinghoff, M. (2008). Combinatorics and graph theory (undergraduate texts in mathematics). *Springer; 2 edition (19 Sep 2008)*.

[48] Hayward, S. (1999). Structural principles governing domain motions in proteins. *Proteins*, 36:425–435.

[49] Hayward, S. and Berendsen, H. (1998). Systematic analysis of domain motions in proteins from conformational change: new results on citrate synthase and t4 lysozyme. *Proteins*, 30:144–145.

[50] Hayward, S., Kitao, A., and Berendsen, H. (1997). Model-free methods of analyzing domain motions in proteins from simulation: A comparison of normal mode analysis and molecular dynamics simulation of lysozyme. *Proteins: Structure, Function, and Bioinformatics*, 27(3):425–437.

[51] Hayward, S. and Lee, R. (2002). Improvements in the analysis of domain motions in proteins from conformational change: DynDom version 1.50. *J Mol Graph Model*, 21:181–183.

[52] Hendrickson, W. (1998). Diffraction and fourier transforms http://www.sci.sdsu.edu/TFrey/bio750/bio750x-ray.html.

[53] Hill, E. (2003). *SCOP Intra-Family Relationships http://compbio.berkeley.edu/people/emma/scop_work.html*.

[54] Hinsen, K. (1998). Analysis of domain motions by approximate normal mode calculations. *PROTEINS: Structure, Function, and Genetics*, 33:417–429.

[55] Holm, L. and Sander, C. (1993). Protein structure comparison by alignment of distance matrices. *J Mol Biol*, 233(1):123–138.

[56] Holm, L. and Sander, C. (1994). Parser for protein folding units. *Proteins*, 19(3):256–268.

[57] Holm, L. and Sander, C. (1996a). The FSSP database: Fold classification based on structure-structure alignment of proteins. *Nucl. Acids Res*, 24(1):206–209.

[58] Holm, L. and Sander, C. (1996b). The FSSP database of structurally aligned protein fold families. *Nucleic Acids Res*, 22(17):3600–3609.

[59] Holm, L. and Sander, C. (1998). Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics*, 14(5):423–429.

[60] Hosmer, D. and Lemeshow, S. (2000). Applied logistic regression (2nd ed.). *Wiley-Liss, Inc.*, ISBN 0-471-35632-8.

[61] Islam, S., Luo, J., and Sternberg, M. (1995). Identification and analysis of domains in proteins. *Protein Eng.*, 8(6):513–526.

[62] Jardetzky, O. (1996). Protein dynamics and conformational transitions in allosteric proteins. *Prog Biophys Mol Biol*, 65(3):171–219.

[63] Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Cryst*, A32:922–923.

[64] Koch, M. and Waldmann, H. (2005). Protein structure similarity clustering and natural product structure as guiding principles in drug discovery. *Drug Discovery Today*, 10(7):471–483.

[65] Krogh, A., Brown, M., Mian, I., Sjölander, K., and Haussler, D. (1994). Hidden markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235(5):1501–1531.

[66] Krzanowski, W. and Hand, D. (2009). ROC curves for continuous data (chapman & hall/CRC monographs on statistics & applied probability). *Chapman and Hall/CRC; 1 edition (1 Jun 2009)*.

[67] kutztown.edu (2011). Nuclear magnetic resonance spectroscopy http://www.kutztown.edu/acad/chem/instruments_html/nmr.htm.

[68] Land, A. and Doig, A. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520.

[69] Lee, C. (1961). An algorithm for path connection and its applications. *IRE Transactions on Electronic Computers*, 10(3):346–365.

[70] Lee, R. (2003). DynDom database and web application design. Technical report.

[71] Lesk, A. and Fordham, W. (1996). Conservation and variability in the structures of serine proteinases of the chymotrypsin family. *J Mol Biol.*, 258(3):501–537.

[72] L'Huillier, S. (1861). Mémoire sur la polyèdrométrie. *Annales de Mathématiques*, 3:169–189.

[73] LifeSciencesFoundation (2014). http://www.lifesciencesfoundation.org/events-xray_crystallography.html. *info@biotechhistory.org*.

[74] Lipschutz, S. (1998). Schaum's outline of set theory and related topics (schaum's outline series). *Schaum's Outlines; 2 edition (1 Jan 1998)*.

[75] Little, J., Murty, C., Katta, G., Sweeney, D., and Karel, C. (1963). An algorithm for the traveling salesman problem. *Operations Research*, 11(6):972–989.

[76] Lo Conte, L., Brenner, S., Hubbard, T., C., C., and Murzin, A. (2002). SCOP database in 2002: refinements accommodate structural genomics. *Nucleic Acids Research*, 30(1):264–267.

[77] Mackay, J. (2004). Symposium: A celebration of australian science http://www.science.org.au/events/sats/sats2004/mackay.html.

[78] Metz, C. (1978). Basic principles of ROC analysis. *Semin Nucl Med*, 8(4):283–298.

[79] Michie, A., Orengo, C., and Thornton, J. (1996). Analysis of domain structural class using an automated class assignment protocol. *J Mol Biol*, 262(2):168–185.

[80] Moore, E. (1959). The shortest path through a maze. *In Proceedings of the International Symposium on the Theory of Switching*, Harvard University Press:285–292.

[81] Moore, S. (2011). Babe ruth and binary logistic regression how the sultan of swat's stats correlate homers and hall of fame. *Quality Digest*, http://www.qualitydigest.com/inside/quality-insider-article/babe-ruth-and-binary-logistic-regression.html.

[82] Müller, C., Schlauderer, G., Reinstein, J., and Schulz, G. E. (1996). Adenylate kinase motions during catalysis: an energetic counterweight balancing substrate binding. *Structure*, 4(2):147–156.

[83] Murphy, K. (2012). Machine learning: A probabilistic perspective (adaptive computation and machine learning series). *MIT Press (18 Sep 2012)*.

[84] Murzin, A., Brenner, S., Hubbard, T., and Chothia, C. (1995). SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247:536–540.

[85] Nave, R. (2000). http://hyperphysics.phy-astr.gsu.edu/hbase/organic/translation.html.

[86] Nemecz, G. (2000). http://web.campbell.edu/faculty/nemecz/323_lect/proteins/prot_chapter.html.

[87] Nichols, W., Rose, G., Ten Eyck, L., and Zimm, B. (1995). Rigid domains in proteins: an algorithmic approach to their identification. *Proteins*, 23(1):38–48.

[88] Nilges, M. (1996). Structure calculation from NMR data. *Curr Opin Struct Biol*, 6(5):617–623.

[89] Orengo, C., Michie, A., Jones, S., Jones, D., Swindells, M., and Thornton, J. (1997). CATH–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108.

[90] Pampel, F. (2000). Logistic regression: A primer (quantitative applications in the social sciences). *SAGE Publications, Inc; First Edition edition (19 July 2000)*.

[91] Pearl, F., Lee, D., Bray, J., Sillitoe, I., Todd, A., Harrison, A., Thornton, J., and Orengo, C. (2000). Assigning genomic sequences to CATH. *Nucleic Acids Res*, 28(1):277–282.

[92] Pearl, F., Todd, A., Sillitoe, I., Dibley, M., Redfern, O., Lewis, T., Bennett, C., Marsden, R., Grant, A., Lee, D., Akpor, A., Maibaum, M., Harrison, A., Dallman, T., Reeves, G., Diboun, I., Addou, S., Lise, S., Johnston, C., Sillero, A., Thornton, J., and Orengo, C. (2005). The CATH domain structure database and related resources gene3d and DHS provide comprehensive domain family information for genome analysis. *Nucleic Acids Res*, 33(Database Issue):D247–D251.

[93] Petsko, G. and Ringe, D. (2004). *Protein Structure and Function*.

[94] Ponting, C. and Russell, R. (2002). The natural history of protein domains. *Annu Rev Biophys Biomol Struct*, 31:45–71.

[95] Poornam, G., Matsumoto, A., Ishida, H., and Hayward, S. (2009). A method for the analysis of domain movements in large biomolecular complexes. *Proteins*, 76(1):201–212.

[96] Qi, G. and Hayward, S. (2009). Database of ligand-induced domain movements in enzymes. *BMC Struct Biol*, 9(13).

[97] Qi, G., Lee, R., and Hayward, S. (2005). A comprehensive and non-redundant database of protein domain movements. *Bioinformatics*, 21(12):2832–2838.

[98] Rhodes, G. (2000). *Crystallography Made Crystal Clear*.

[99] Richardson, J. (1981). The anatomy and taxonomy of protein structure. *Adv Protein Chem*, 34:167–339.

[100] Robbins, H. (1939). A theorem on graphs, with an application to a problem on traffic control. *American Mathematical Monthly*, 46:281–283.

[101] Rose, G. (1979). Hierarchic organization of domains in globular proteins. *J. Mol. Biol.*, 134(3):447–470.

[102] Rupp, B. (2010a). *About Crystals, Symmetry and Space Groups http://www.ruppweb.org/xray/tutorial/Crystal_sym.html*.

[103] Rupp, B. (2010b). *Crystallography 101 http://www.ruppweb.org/xray/101index.html*.

[104] Russell, R. (2006). *Chemistry 202 Peptide Secondary Structure http://www.nku.edu/russellk/tutorial/peptide/peptide.html*.

[105] Sander, C. and Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, 9(1):56–68.

[106] Sander, C. and Schneider, R. (1994). The HSSP database of protein structure-sequence alignments. *Nucleic Acids Res.*, 22:3597–3599.

[107] Sayle, R. and Milner-White, E. (1995). RASMOL: biomolecular graphics for all. *Trends Biochem Sci*, 20(9):374–376.

[108] Schneider, R., de Daruvar, A., and Sander, C. (1997). The HSSP database of protein structure-sequence alignments. *Nucleic Acids Res*, 25:226–230.

[109] Schneider, R. and Sander, C. (1996). The HSSP database of protein structure-sequence alignments. *Nucleic Acids Res*, 24:201–205.

[110] Schultz, J., Milpetz, F., Bork, P., and Ponting, C. (1998). SMART, a simple modular architecture research tool: identification of signaling domains. *Proc Natl Acad Sci USA*, 95(11):5857–5864.

[111] Schwartz, R. (1969). Invariant proper bayes tests for exponential families. *Ann. Math. Statist*, 40(1):270–283.

[112] Servant, F., Bru, C., Carrère, S., Courcelle, E., Gouzy, J., Peyruc, D., and Kahn, D. (2002). ProDom: automated clustering of homologous domains. *Brief Bioinform*, 3(3):246–251.

[113] Siddiqui, A. and Barton, G. (1995). Continuous and discontinuous domains: An algorithm for the automatic generation of reliable protein domain definitions. *Protein Science*, 4(5):872–884.

[114] Sinha, N., Kumar, S., and Nussinov, R. (2001). Interdomain interactions in hinge bending transitions. *Structure*, 9:1165–1181.

[115] Sonnhammer, E. and Kahn, D. (1994). Modular arrangement of proteins as inferred from analysis of homology. *Protein Sci*, 3(3):482–492.

[116] Stedman, T. (2004). *The American Heritage Stedman's medical dictionary*.

[117] Steen, M. (2010). Graph theory and complex networks: An introduction. *Maarten van Steen (5 April 2010)*.

[118] Steinmetz, M. and Akhmanova, A. (2008). Capturing protein tails by CAP-gly domains. *Trends in Biochemical Sciences*, 33(11):535–545.

[119] Stoldt, M., Wöhnert, J., Görlach, M., and Brown, L. (1998). The NMR structure of escherichia coli ribosomal protein l25 shows homology to general stress proteins and glutaminyl-tRNA synthetases. *The EMBO Journal*, 17:6377–6384.

[120] Stoll, R. (2003). Set theory and logic (dover books on mathematics). *Dover Publications Inc.; New edition edition (17 Mar 2003)*.

[121] Swindells, M. (1995). A procedure for detecting structural domains in proteins. *Protein Sci*, 4(1):103–112.

[122] Sylvester, J. (1878). Chemistry and algebra. *Nature*, 17:284.

[123] Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160.

[124] Teague, S. (2003). Implications of protein flexibility for drug discovery. *Nature Reviews Drug Discovery*, 2:527–541.

[125] Tetko, I., Livingstone, D., and Luik, A. (1995). Neural network studies. 1. comparison of overfitting and overtraining. *J. Chem. Inf. Comput. Sci.*, 35(5):826–833.

[126] Theobald, D. (2012). 29+ evidences for macroevolution. *Phylogenetics Primer*, http://www.talkorigins.org/faqs/comdesc/phylo.html#fig2.

[127] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser*, 1:267–288.

[128] Till, D. and Hand, R. (2012). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45:171–186.

[129] Tremaux, C. (1876). Maze solving algorithm. *Ecole Polytechnique of Paris*.

[130] Trudeau, R. (2003). Introduction to graph theory (dover books on mathematics). *Dover Publications Inc.; 2nd Revised edition edition (17 Mar 2003)*.

[131] Wernisch, L., S. J. (2003). Structual bioinformatics (chapter 18 identifying stryctual domains in protein).

[132] Wikipedia (2010). *http://en.wikipedia.org/wiki/File:Receiver_Operating_Characteristic.png*.

[133] Wiley, J. (2004). *http://www.uic.edu/classes/bios/bios100/lecturesf04am/lect02.html*.

[134] Wilson, R. (2010). Introduction to graph theory. *Prentice Hall; 5 edition (20 May 2010)*.

[135] Wodak, S. and Janin, J. (1981). Location of structural domains in proteins. *Biochemistry*, 20(23):6544–6552.

[136] Wriggers, W., Chakravarty, S., and Jennings, P. (2005). Control of protein functional dynamics by peptide linkers. *Biopolymers*, 80(6):736–746.

[137] Wriggers, W. and Schulten, K. (1997). Protein domain movements: Detection of rigid domains and visualization of effective rotations in comparisons of atomic coordinates. *Proteins: Structure, Function, and Genetics*, 29:1–14.

[138] Yeats, C. and Orengo, C. (2007). Evolution of protein domains. *ENCYCLOPEDIA OF LIFE SCIENCES*.

[139] Zheng, S., Song, R., Wen, J., and Wu, D. (2008). Joint optimization of wrapper generation and template detection. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 894–902.

[140] Zou, K., Liu, A., Bandos, A., Ohno-Machado, L., and Rockette, H. (2011). Statistical evaluation of diagnostic performance: Topics in ROC analysis (chapman & hall/CRC biostatistics series). *Chapman and Hall/CRC (12 Aug 2011)*.

[141] Zweig, M. and Campbell, G. (1993). Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. *Clin Chem*, 39(4):561–577.

Appendix A

Training Set for Logistic Regression

Protein Name	DBMM Assignment	PDB 1	Chain ID 1	PDB 2	Chain ID 2	Rotation Angle (deg)	Prediction Value	Predicted Class
Alcohol Dehydrogenase (ADH)	Predominantly Shear	1N8K	A	1YE3	A	8.5	0.171	Hinge
Aspartate Amino Transferase (AAT)	Predominantly Shear	1AKB	A	7AAT	B	13.3	0.314	Hinge
Calpain protease core	Predominantly Shear	2G8J	A	1TL9	A	13.4	0.319	Hinge
Citrate Synthase	Predominantly Shear	1CTS	null	1CSH	null	19.4	0.547	Mixed
Glyceraldehyde-3-phosphate Dehydrogenase	Predominantly Shear	2GD1	R	1NQ5	A	8.3	0.443	Hinge
SARS virus protease	Predominantly Shear	1Z1J	A	1UK2	A	13.9	0.563	Shear
Trp Repressor (TrpR)	Predominantly Shear	1ZT9	D	1WRP	R	12.1	0.647	Shear
Actin	Predominantly Shear	1HLU	A	1MDU	E	13.5	0.164	Hinge
Aspartyl tRNA Synthetase	Predominantly Shear	1G51	B	1EFW	A	11.4	0.426	Hinge
Cytochrome P450BM-3	Predominantly Shear	1JPZ	B	1BVY	A	15.1	0.759	Shear
DNA Polymerase III	Predominantly Shear	1MMI	A	1JQL	A	13.7	0.375	Hinge
E. coli clamp loader gamma subunit	Predominantly Shear	1JR3	D	1XXH	A	18.3	0.591	Shear
E. Coli Mta/Adohcy Nucleosidase	Predominantly Shear	1NC1	A	1JYS	B	8.6	0.622	Shear
Endothiapepsin	Predominantly Shear	1GVU	A	4APE	null	3.8	0.999	Shear
Glutamyl tRNA synthetase	Predominantly Shear	1N78	B	1GLN	null	10.9	0.67	Shear
Heat Shock Protein (HSP)	Predominantly Shear	1HX1	A	1KAZ	null	12.7	0.265	Hinge
Hexokinase	Predominantly Shear	2YHX	A	1HKG	A	13	0.789	Shear
Molybdate-binding protein	Predominately Shear	1HK9	A	1H9M	A	8.5	0.311	Hinge
Phenylalanine Hydroxylase	Predominantly Shear	1MMK	A	1J8U	A	16.1	0.27	Hinge
Phosphofructokinase (PFK) (not allosteric transition)	Predominantly Shear	1PFK	A	1PFK	B	5.3	0.864	Shear
PvuII endonuclease	Predominantly Shear	1NI0	B	3PVI	A	33.3	0.411	Hinge
Threonine tRNA Synthetase	Predominantly Shear	1EVK	A	1EVL	D	11.4	0.441	Hinge
Tyrosine Kinase-Type Cell Surface Receptor Her2	Predominantly Shear	1N8Z	B	2FJG	B	20.3	0.411	Hinge
ATP Sulfurylase	Predominantly Hinge	1I2D	C	1M8P	A	21	0.167	Hinge

cAMP-dependent Protein Kinase (catalytic domain)	Predominantly Hinge	1JLU	E	1CMK	E	13.4	0.361	Hinge
c-Src tyrosine kinase	Predominantly Hinge	1YI6	B	1FMK	A	22.1	0.314	Hinge
Folylpolyglutamate Synthetase	Predominantly Hinge	1JBW	A	2GC5	A	15.4	0.132	Hinge
HCV Helicase	Predominantly Hinge	8OHM	null	1CU1	B	35.3	0.063	Hinge
T7 Phage RNA Polymerase	Predominantly Hinge	1ARO	P	1CEZ	A	11.8	0.751	Shear
Thioredoxin reductase/ Glutathione reductase	Predominantly Hinge	1F6M	B	1TRB	null	65.8	0.212	Hinge
Transferrins (N-terminal lobe)	Predominantly Hinge	1RYO	A	1BP5	C	62.9	0.003	Hinge
Troponin-C	Predominantly Hinge	1YTZ	C	1TOP	null	47.8	0.029	Hinge
Uracil-DNA Glycosylase	Predominantly Hinge	1EMH	A	2HXM	A	6.7	0.895	Shear
3-Isopropylmalate Dehydrogenase	Predominantly Hinge	1OSJ	A	1IDM	null	15.3	0.253	Hinge
Acetylcholinesterase	Predominantly Hinge	2CMF	A	2J4F	A	5.2	0.525	Mixed
Acetyl-CoA synthase	Predominantly Hinge	1OAO	D	1MJG	N	52.1	0.01	Hinge
Adenylate Kinase (ADK)	Predominantly Hinge	1E4V	A	1E4Y	B	15.7	0.27	Hinge
Arabinose, Leucine, and Galactose Binding Proteins	Predominantly Hinge	2FW0	A	2HPH	A	35.4	0.137	Hinge
Biotin carboxylase	Predominantly Hinge	1BNC	A	1DV2	A	47.1	0.195	Hinge
C. Glutamicum DAP Dehydrogenase	Predominantly Hinge	1F06	B	2DAP	null	10.7	0.079	Hinge
Calmodulin	Predominantly Hinge	1QX5	J	1QX7	R	13.7	0.282	Hinge
Catabolite Gene Activator Protein (CAP)	Predominantly Hinge	1ZRE	B	1O3Q	A	12	0.314	Hinge
CBL	Predominantly Hinge	1B47	A	1YVH	A	16.1	0.319	Hinge
Cell Adhesion Molecule CD2	Predominantly Hinge	1CDC	A	1A64	A	97.7	0.113	Hinge
Cyanovirin-N	Predominantly Hinge	1L5B	B	3EZM	A	72	0.411	Hinge
Diphtheria Toxin (DT)	Predominantly Hinge	1F0L	B	1TOX	B	177.4	0	Hinge
DNA Beta-Glucosyltransferase	Predominantly Hinge	1JEJ	A	1M5R	A	14.8	0.162	Hinge
DNA Polymerase Beta (Pol Beta)	Predominantly Hinge	2FMP	A	7ICO	A	37.1	0.314	Hinge
E. coli. Periplasmic Dipeptide Binding Protein	Predominantly Hinge	1DPE	null	1DPP	A	53.7	0.001	Hinge
Elongation Factor G	Predominantly Hinge	2EFG	A	1FNM	A	13.2	0.418	Hinge

Eukaryotic RNA Polymerase	Predominantly Hinge	1I50	A	1I6H	A	30.1	0.223	Hinge
Ferric binding protein	Predominantly Hinge	1MRP	null	1NNF	A	21.9	0.153	Hinge
Formate Dehydrogenase (FDH)	Predominantly Hinge	2NAC	A	2NAD	A	8.1	0.265	Hinge
Glur2 ligand-binding core	Predominantly Hinge	2I3V	A	2CMO	A	26	0.113	Hinge
Glutamate Dehydrogenase	Predominantly Hinge	1AUP	null	1HRD	C	24.7	0.015	Hinge
Glutamine Binding Protein	Predominantly Hinge	1GGG	A	1WDN	A	55.7	0.052	Hinge
Glutaminyl-tRNA synthase	Predominantly Hinge	1GTR	A	1NYL	A	9.6	0.47	Mixed
Glycerate Dehydrogenase (GDH)	Predominantly Hinge	1PSD	A	1YBA	D	12.5	0.47	Mixed
GroEL domain	Predominantly Hinge	1AON	G	1XCK	N	84.5	0.164	Hinge
Guanylate Kinase	Predominantly Hinge	1EX6	B	1EX7	A	47	0.361	Hinge
Kinesin-like KIF1A Motor Domain	Predominantly Hinge	1I5S	A	1VFV	A	23.1	0.622	Shear
Lactoferrin	Predominantly Hinge	1CB6	A	1LCF	null	55.2	0.015	Hinge
Lysine/Arginine/Ornithine (LAO) binding protein	Predominantly Hinge	2LAO	null	1LST	n	51	0.063	Hinge
Maltodextrin Binding Protein (MBP)	Predominantly Hinge	1OMP	A	3MBP	A	34.8	0.034	Hinge
Methylene-Tetrahydromethanopterin Dehydrogen	Predominantly Hinge	1LU9	A	1LUA	A	8.3	0.536	Mixed
mRNA capping enzyme	Predominantly Hinge	1CKO	A	1CKM	B	31.9	0.137	Hinge
Mura (Udp-N-Acetylglucosamine Enolpyruvyltransferase	Predominantly Hinge	1UAE	null	1EJD	B	17.2	0.02	Hinge
Oligopeptide-binding protein	Predominantly Hinge	1RKM	null	1JET	A	25.7	0.008	Hinge
Phosphate-binding protein	Predominantly Hinge	1QUK	A	1OIB	A	26.2	0.094	Hinge
Phosphoglycerate Kinase	Predominantly Hinge	13PK	A	1PHP	A	27.1	0.025	Hinge
Replication protein A DNA-binding domain	Predominantly Hinge	1FGU	B	1JMC	A	96.2	0.019	Hinge
Ribose Binding Protein	Predominantly Hinge	1BA2	A	2DRI	null	62.9	0.028	Hinge
Ribose-5-Phosphate Isomerase	Predominantly Hinge	1KS2	A	1KS2	B	12.1	0.439	Hinge
T4 lysozyme mutants: Ile3 to Pro & Met6 to Ile	Predominantly Hinge	1L96	A	1L97	A	31	0.411	Hinge
Tryptophan Synthase	Predominantly Hinge	2TYS	B	1QOQ	B	14.9	0.012	Hinge
Type-C Inorganic Pyrophosphatase	Predominantly Hinge	1K20	B	1WPP	A	19.2	0.023	Hinge
Various Kinases (Tyr, Ser, Thr)	Predominantly Hinge	2FYS	A	2OJG	A	48.1	0.124	Hinge

Appendix B

Published Research Papers

OPEN ACCESS Freely available online

PLOS ONE

Classification of Domain Movements in Proteins Using Dynamic Contact Graphs

Daniel Taylor, Gavin Cawley, Steven Hayward*

D'Arcy Thompson Centre for Computational Biology, School of Computing Sciences, University of East Anglia, Norwich, United Kingdom

Abstract

A new method for the classification of domain movements in proteins is described and applied to 1822 pairs of structures from the Protein Data Bank that represent a domain movement in two-domain proteins. The method is based on changes in contacts between residues from the two domains in moving from one conformation to the other. We argue that there are five types of elemental contact changes and that these relate to five model domain movements called: "free", "open-closed", "anchored", "sliding-twist", and "see-saw." A directed graph is introduced called the "Dynamic Contact Graph" which represents the contact changes in a domain movement. In many cases a graph, or part of a graph, provides a clear visual metaphor for the movement it represents and is a motif that can be easily recognised. The Dynamic Contact Graphs are often comprised of disconnected subgraphs indicating independent regions which may play different roles in the domain movement. The Dynamic Contact Graph for each domain movement is decomposed into elemental Dynamic Contact Graphs, those that represent elemental contact changes, allowing us to count the number of instances of each type of elemental contact change in the domain movement. This naturally leads to sixteen classes into which the 1822 domain movements are classified.

Citation: Taylor D, Cawley G, Hayward S (2013) Classification of Domain Movements in Proteins Using Dynamic Contact Graphs. PLoS ONE 8(11): e81224. doi:10.1371/journal.pone.0081224

Editor: Danilo Roccatano, Jacobs University Bremen, Germany

Received July 31, 2013; **Accepted** October 9, 2013; **Published** November 18, 2013

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: sjh@cmp.uea.ac.uk

Introduction

From a structural perspective domains in proteins can be regarded as quasi-globular regions. The connections between domains allow their relative movement and consequently domain movements are often engaged in protein function [1,2]. The Protein Data Bank (PDB) [3] is a rich source of information on protein domain movements as for a number of proteins, multiple structures have been deposited. Differences in structure may be due to functional changes in state as occurs upon the binding of a natural ligand, but may also be due to differences in the experimental conditions under which the structures were solved, or could be due to natural or engineered mutations. The implied movements between multiple structures of certain proteins deposited in the PDB invite a computational biology approach in order to understand principles and causes of protein conformational change. For domain proteins there have been a number of such studies. As understanding in biology often follows classification of experimental findings some of these studies have attempted to classify the implied movements in domain proteins.

In an influential review of protein domain movements using structures from the PDB, Gerstein et al. [4] saw two main types: predominantly hinge and predominantly shear. Following this study the DataBase of Macromolecular Movements (DBMM) appeared online with further examples [5]. A number of other large-scale studies have been made using structures from the PDB each approaching the problem from a different perspective. A study of movements in enzymes upon substrate binding reported that they are generally small [6], although another study has shown that the extent of movement may depend on the actual reaction mechanism [7]. A study based on the DynDom program [8,9] for the analysis of domain movements in proteins considered structural features of hinge-bending regions [10] and the application of the same program to create a Non-redundant DataBase of Protein Domain Movements (NRDPDM) showed that protein domain movements are very controlled in the sense that many different structures from the same family represent the same domain movement [11]. The "Database of Ligand-Induced Domain Movements in Enzymes," [12] which is a subset of the NRDPDM, categorised domain movements in 203 enzymes based on whether a ligand "spans" the two domains or not and whether the ligand has caused compaction of the proteins upon binding. A more general approach has been taken to produce the Protein Structural Change DataBase (PSCDB) [13,14] where 839 protein movements between liganded and unliganded structures have been classified into seven categories: "coupled domain motion", "independent domain motion", "coupled local motion", "independent local motion", "burying ligand motion", "no significant motion", and "other type of motion". Related to these studies is another large scale study which considered 521 structural pairs with the conformational change apparently induced by ligand binding [15]. Although this study did not classify domain movements it did consider the predictability of domain movements from the ligand-free form. Another way to approach the subject of domain movements in proteins is to consider the energetics of the process. Sinha et al. [16] showed that for a number of domain proteins the nonpolar buried surface area in the open state

matches or exceeds the nonpolar buried surface area in the closed state.

The method presented here is based on changes in interdomain residue contacts that occur in the domain movement. The advantage of such a method is that it is relatively simple to implement but has a connection to methods based on calculating interaction energies. Key to the analysis is the concept of the "Dynamic Contact Graph" (DCG). Each domain movement has an associated DCG. Using graphs has three benefits: they provide a visual metaphor for the movement they represent; they provide motifs for some basic domain movements that are instantly recognisable; the well-developed algorithms of graph theory can be used to evaluate features of interest. The analysis is developed in terms of "elemental" DCGs which represent elemental contact-changes. These elemental contact-changes can, under certain assumptions, be associated with "model" domain movements. We count the number of elemental DCGs any general DCG comprises which naturally leads to sixteen different categories into which the domain movements are classified. The results are presented at a website.

Methods

Database

The basic data are the 2035 unique domain movements from the NRDPDM [11]. The domain movements were determined by the DynDom program[8,9]. These unique movements come from 1578 families which means that some domain movements are from the same family. Individual cases from this dataset are available to browse at http://www.cmp.uea.ac.uk/dyndom. In order to simplify the analysis only those cases with two domains were used. Of the 2035 cases, 1822 are two-domain proteins. The two domains in each protein will be referred to as "domain A" and "domain B" below.

Residue contact definition

"Contact" between residue i and residue j means any heavy atom of residue i is within 4 Å of any heavy atom of residue j. However, before the set of pair-wise contacts between residues in each domain and for each conformation is determined, residues at the boundaries of the domains annotated by DynDom as bending regions were removed as were residues close to the interdomain screw axis (any heavy atom of the residue within 5.5 Å of the axis). The reason for this is that they would be expected to have maintained contacts (see below) irrespective of the nature of the domain movement.

Elemental contact-changes and model domain movements

Let (a_1,b_1) denote a "residue contact pair", where a_1 is the residue number of a residue in domain A, and b_1 is the residue number of a residue in domain B, that make contact in conformation 1. Similarly let (a_2,b_2) represent a residue contact pair in conformation 2. By considering *at most* a single residue contact pair between the domains in *either* conformation there are five "elemental contact-change" scenarios (where below $()$ indicates no contact exists):

- "**no-contact**": $(a_1,b_1) = ()$ and $(a_2,b_2) = ()$.
- "**new**": either $(a_1,b_1) \neq ()$ and $(a_2,b_2) = ()$ or $(a_1,b_1) = ()$ and $(a_2,b_2) \neq ()$.
- "**maintained**": $(a_1,b_1) \neq ()$ and $(a_2,b_2) \neq ()$ where $a_1 = a_2$ and $b_1 = b_2$.
- "**exchanged-partner**": $(a_1,b_1) \neq ()$ and $(a_2,b_2) \neq ()$ where $(a_1 = a_2$ and $b_1 \neq b_2)$ or $(a_1 \neq a_2$ and $b_1 = b_2)$.
- "**exchanged-pair**": $(a_1,b_1) \neq ()$ and $(a_2,b_2) \neq ()$ where $a_1 \neq a_2$ and $b_1 \neq b_2$.

The contact-changes can be associated with five "model" domain movements assuming the following idealisation.

- The domains have a spherical shape and are perfectly rigid.
- There is only one residue from each domain at a contact point.
- The relative movement of the domains is a rotation about a hinge axis passing through an interdomain linker region which is short in comparison to the size of the domains.

The "no contact" case implies the domains remain separated and can move freely. This case we call "*free*". The "new" case implies the domains move from a contacting to non-contacting conformation (or vice-versa) suggesting a rotation about a hinge axis perpendicular to the line joining the centres of mass of the domains, defined previously as a "closure" motion[17]. This is called an "*open-closed*" movement. The "maintained" case means the domains cannot move (given that we exclude the hinge region which would otherwise be designated as maintained region) implying the domains remain "*anchored*". For the "exchanged-partner" case we have the same residue from one domain making a contact in both conformations but with different residues on the other domain. This implies one domain sliding over the other and is easiest to imagine occurring by a relative twist of the domains. Consider the hinge axis passing through the centre of mass of domain A, with the centre of mass of domain B slightly shifted from the hinge axis, i.e. predominantly a twist motion [17]. If contact occurs between the two domains then the contact point (residue) on domain B will trace out a circle on domain A. So, residue B will contact two different points (residues) on domain A in a movement. We call this movement a "*sliding-twist*". For the "exchanged-pair" case, the two residues making contact in one conformation are not involved in making contact in the other conformation again implying a movement with the hinge axis perpendicular to the line joining the centres of mass. The movement would break the contact on one side of the domains and rotation continues until contact is made on the other side of the domains. This is commonly known as a "*see-saw*" motion which has already been seen to occur in lactoferrin [18]. More realistic interpretations of these five model domain movements with non-spherical domains and residues of finite size are illustrated in Figure 1.

The association of these elemental contact-changes with the model domain movements is based on consideration of the simplest, most plausible domain movement to reproduce the elemental contact-change in an idealised system. In reality even in those cases where only one type of elemental contact-change occurs, the movement might not resemble the corresponding model domain movement as domains are not perfectly rigid and often have complex interfaces. The extent to which real domain movements conform to these idealised movements is something to be determined.

Dynamic Contact Graphs

Here we introduce Dynamic Contact Graphs (DCGs). Let $\{(a_{1i},b_{1i})\}$, $i = 1,N_1$ denote the set of residue contact pairs in conformation 1 and $\{(a_{2i},b_{2i})\}$, $i = 1,N_2$ the corresponding set for conformation 2.

Each node of the graph represents a residue of which there are two types: those in domain A and those in domain B. An edge

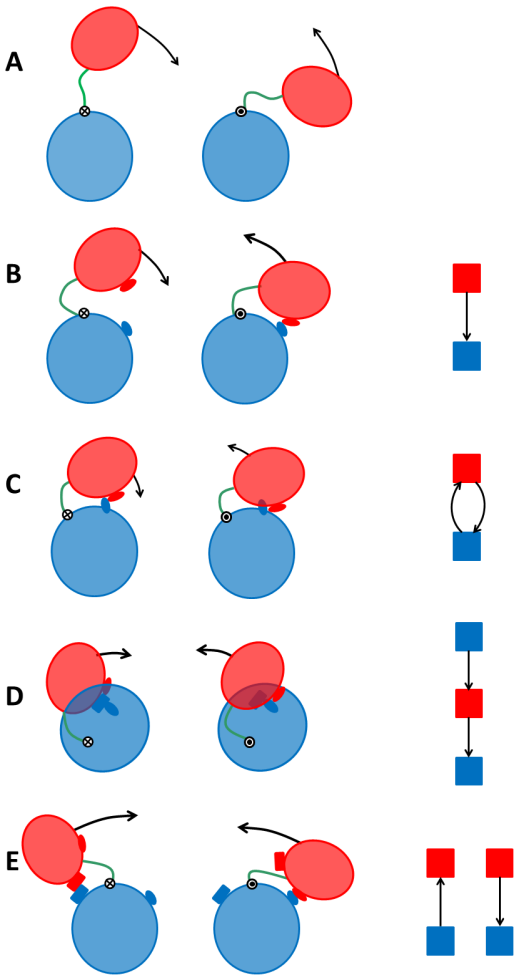

Figure 1. The five model domain movements and their corresponding elemental DCGs. Conformation 1 is on the left and conformation 2 on the right with domain A in blue and domain B in red. (A) The "no contact" contact-change implies that the domains are "free" to move. The graph is empty in this case. (B) The "new" contact-change implies an "open-closed" domain movement. In this case the elemental DCG shows a contact between the two domains in conformation 2 as indicated by the edge-arrow pointing from domain B to domain A. (C) The "maintained" case implies the domains are "anchored" and the associated DCG is a doubly-linked motif. (D) The "exchange-partner" contact-change is where a residue, here on domain B, makes a contact with a residue on domain A in conformation 1 and a contact with a different residue on domain A in conformation 2. This implies a model "sliding-twist" movement whereby domain B slides on the surface provided by domain A. The elemental DCG provides a visual metaphor for this movement with arrows indicating a movement away the contacting residue on domain A in conformation 1 (upper blue node) towards the contacting residue on domain A in conformation 2 (lower blue node). (E) The "exchanged-pair" contact-change and its associated model "see-saw" movement. The DCG clearly depicts this kind of see-saw movement.
doi:10.1371/journal.pone.0081224.g001

exists when there is a contact between a residue in domain A and a residue in domain B, i.e. when they appear in one of the sets above. The key feature of a DCG is that it is directed. For contacts in conformation 1 the direction associated with an edge is from the residue (node) in domain A to the residue (node) in domain B (call this an AB edge). This could be written as $a_{1i} \rightarrow b_{1i}$. For contacts in conformation 2 the direction is from the residue (node) in domain B to the residue (node) in domain A (call this a BA edge). This could be written as $a_{2i} \leftarrow b_{2i}$. Figure 1 shows the "elemental DCGs" for the five model domain movements.

In general a domain movement may combine these elemental contact-changes and have a complex graph structure.

We make full use of Matlab (version 8.0.0.783 (R2012b)) and in particular the Bioinformatics Toolbox "biograph" function to create a "biograph" object, a data structure for directed graphs. This enabled us to use associated methods to analyse and view the DCGs.

Results

Information on each domain movement can be found at our website. Each domain movement has its own webpage on which its DCG is shown. However, 413 domain movements have no contacts in both conformations (apart from at the removed hinge regions). For these the DCG is empty. These domain movements are assigned to the "no contact" class which implies a free movement of the domains.

The remaining 1409 domain movements each have a DCG. An illustrative example from a DNA topoisomerase III is shown in Figure 2. Our aim is to process each DCG in order to count how many of each of the four elemental contact-changes are contained within it (we ignore no contact which applies to all the residues not contained in the graph and is only interesting when all residues in the protein are in this category). The distribution of the number of instances of each of the four elemental contact-changes in each DCG will allow us to classify the domain movements.

Decomposing DCGs into the elemental contact-changes - principles

As can be seen in Figure 2, DCGs are not necessarily connected. A disconnected graph means that residues in one subgraph do not make contact with any residues from another disconnected subgraph in either conformation, indicating independent regions that are possibly playing a different role in the domain movement. We use the Matlab Bioinformatics Toolbox's "biograph" object method "conncomp" to count the number of disconnected subgraphs for all DCGs. This information is presented on the webpage of each domain movement.

Our aim is to count the number of contact-changes of each type for each domain movement. This is equivalent to decomposing a DCG into the four elemental DCGs shown in Figure 1. Identifying a contact change implies that a pair of contacts in one conformation have to be associated with a pair of contacts (or indeed lost contacts) in the other conformation. Identifying and counting maintained contact-changes (which appear as double links in the graph) is an unambiguous process. Let N_{maint} represent the number of maintained-changes. For the DNA topoisomerase III shown in Figure 2 $N_{maint} = 7$. Counting exchanged-partner contact-changes is not unambiguous as illustrated in Figure 3. In Figure 3A there is a single contact between residues 1 and 4 in conformation 1, but after a sliding movement there are two contacts in conformation 2. The ambiguity lies in whether it is residue 1 that exchanges contact partner 4 with 3, or whether it is residue 4 that exchanges contact partner 1 with 2. In the DCG this is equivalent to identifying the elemental DCGs for an exchanged-partner contact-change which is a triplet (three nodes connected by two edges with the same direction). In this example we can select the triplet 3-1-4 or the triplet 1-4-2. Note that we cannot count both as we are counting types of contact-changes and counting both would mean that the 1-4 contact is counted twice. If we select the triplet 3-1-4 then the new contact is 2-4; if we select

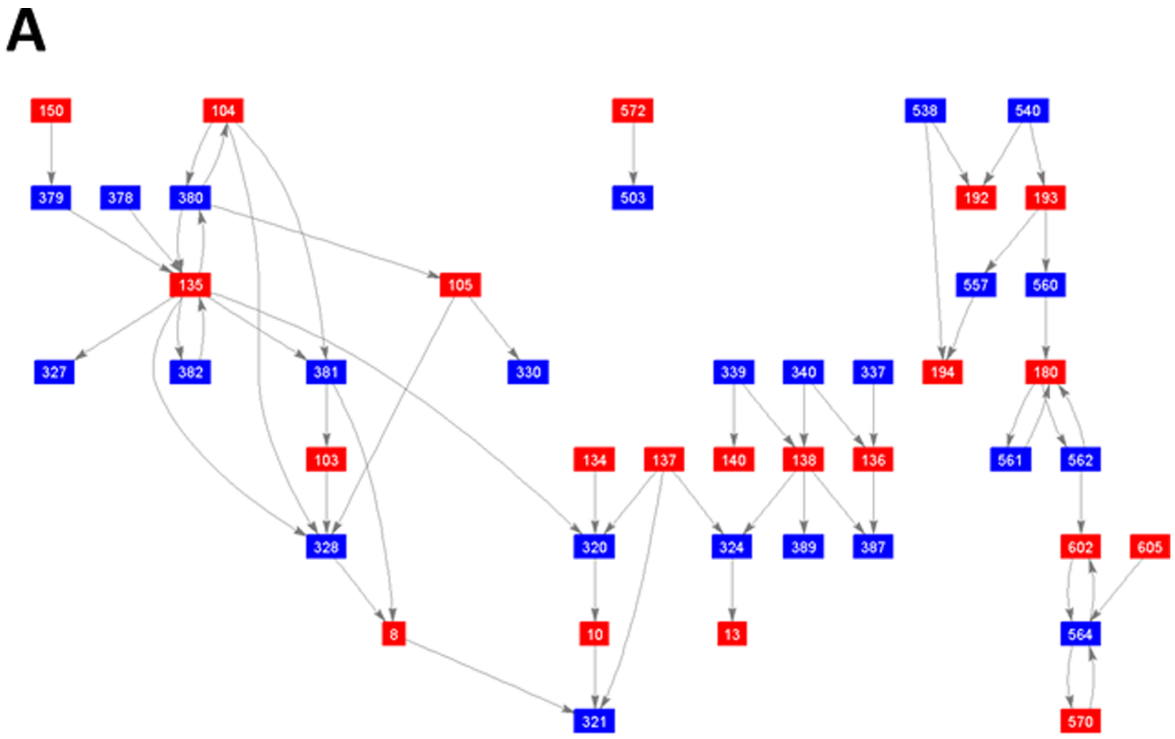

Figure 2. DCG and bar chart for DNA topoisomerase III. (A) DCG for DNA topoisomerase III for the movement between structural pair: 1I7D, chain A, and 1D6M, chain A. (B) Decomposition of the DCG determines the number of instances in each of the four types of elemental contact-changes, ''maintained'', ''exchanged-partner'', ''exchanged-pair'' and ''new'', which are displayed in a bar chart.
doi:10.1371/journal.pone.0081224.g002

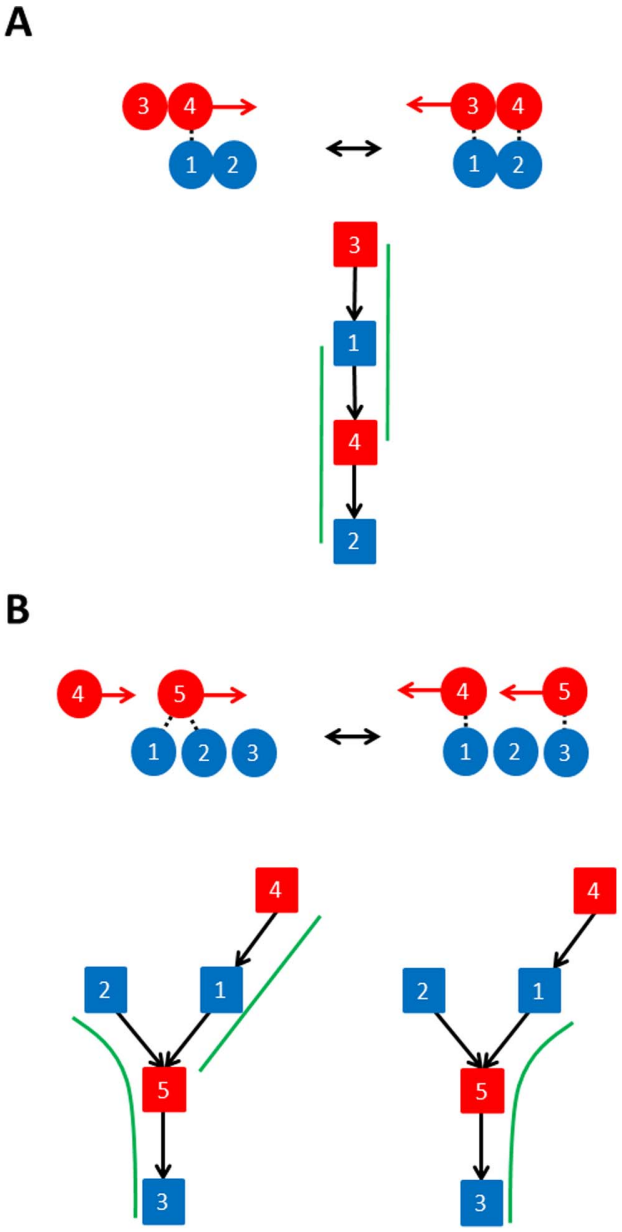

Figure 3. Illustrations of the ambiguity in decomposing a DCG into the elemental "exchange-partner" DCGs. Filled circles indicate residues, those coloured blue are from domain A and those coloured red from domain B. A contact is indicated by a broken line. (A) Top: residues 3 and 4 on domain B slide on residues 1 and 2 on domain A. This can be interpreted as either residue 4 sliding on the surface provided by 1 and 2 or residue 1 sliding on the surface provided by 3 and 4. Bottom: for the associated DCG the elemental "exchange-partner" DCGs are indicated by the green lines but only one can be selected as they should not overlap. (B) Top: residues 4 and 5 on domain B slide on residues 1, 2 and 3 on domain A. Bottom: there are two decomposition possibilities of the DCG indicated by the green lines, one with two non-overlapping elemental "exchange-partner" DCGs (left), and the other with one non-overlapping elemental "exchange-partner" DCG (right).
doi:10.1371/journal.pone.0081224.g003

the triplet 1-4-2 then the new contact is 1-3 and in the absence of any further information both are valid. In practice only one will be selected (see below). This example shows that for exchanged-partner contact-changes we should select only non-overlapping triplets in a DCG.

Figure 3B illustrates another example where there are two possible solutions. One solution has two exchanged-partner contact-changes: residue 1 (in domain A) slides on the surface of residues 4 and 5 (in domain B), and residue 5 (in domain B) slides on the surface of residues 2 and 3 (in domain A). The other solution gives just one exchanged-partner contact-change: residue 5 (in domain B) slides on the surface provided by residues 1 and 3 (in domain A). If we choose the latter then the interactions between residues 2 and 5 in conformation 1 and 1 and 4 in conformation 2 would be assigned to an exchanged-pair contact-change, indicating a possible see-saw movement. In terms of the DCG one can easily see that there are two possible ways to fit non-overlapping triplets in this graph, one gives one triplet, the other, two triplets. How do we in the absence of any other information decide which one to select? Although both are possible, it is more likely that given some of the residues are in an exchange-partner contact-change indicating a sliding movement then all residues would be sliding and therefore in an exchanged-partner contact-change. Therefore we should maximise the number of exchanged-partner contact-changes in a graph. An alternative argument would be that we should maximise the number of associated contact pairings in a graph (in an exchanged-partner contact-change two contact pairs one from each conformation are associated via the residue that appears in both) before pairing off contact pairs to the exchanged-pair contact-changes for which there is no association.

The problem of identifying exchanged-partner contact-changes is therefore equivalent to finding the maximum number of non-overlapping triplets in the DCG.

Once the maintained and exchanged-partner contact-changes have been assigned the exchanged-pair and new contact-changes are assigned as detailed below.

Decomposing DCGs into the elemental contact-changes - practice

The first step counts the number of maintained contact-changes in a DCG and then creates a new DCG that has no double links. The maximum number of non-overlapping triplets in the resulting graph was then determined as follows. First all possible triplets (overlapping and non-overlapping) were determined. A new (undirected) graph was then created which had a node (vertex) for each triplet and an edge between any two nodes with triplets that overlap. An exhaustive search was implemented to find the maximum number of non-overlapping triplets. The algorithm involved selecting a node, removing those nodes connected with it by a single edge and repeating this process until no nodes remain. The selected nodes give a set of non-overlapping triplets. This recursive program is given here in pseudo-code:

Input: A graph with vertices (nodes, representing triplets) ordered, $V = v_1, v_2, v_3, \ldots, v_n$ and a set of edges E (an edge existing if the two vertices represent triplets that overlap).

Output: A list of vertices, W_{max}, with the maximum number of vertices, N_{max}, none of which are connected by a single edge.

$N_{max} = 0$
$W_{max} = \{\}$
$W = \{\}$
add v_1 to W
$w = v_1$
$V' = V$
unconnected$(w, V', E, W, W_{max}, N_{max})$ {

```
if (|V′| = 0){
if (|W| > N_max) {
W_max = W
N_max = |W|
}
return W_max, N_max
# terminate branch in search tree if it cannot
# exceed N_max
} elseif (|V′|+|W| <= N_max){
return
}
while (there is an edge (w, v_j) ϵ E) {
remove v_j from V′
}
remove w from V′
add v_i to W # v_i appears first in V′
w = v_i
# recursive call to unconnected
unconnected(w, V′, E, W, W_max, N_max)
}
```

For twelve DCGs this exhaustive search was too slow and was replaced by a related random search (Repeat the following N times: randomly select a vertex w, add to W; remove vertices with an edge connecting to w; continue first two steps until exhaustion of vertices. Then search amongst the N W recorded for each repetition for N_{max} and W_{max}). This random search found the same value of N_{max} determined by the exhaustive search in all 1397 DCGs that could be search exhaustively. $N_{exchpart}$, the number of exchanged-partner contact-changes is set equal to N_{max}.

The maximum number of non-overlapping triplets is not a unique set but only one is delivered by the exhaustive search given above. For the purpose of this study it does not matter which set we select as we are interested only in the number of each type of elemental contact-change.

A DCG with maintained and exchanged-partner contact-changes removed comprises disconnected two-node subgraphs. Each subgraph has a single AB edge for conformation 1 or a single BA edge for conformation 2 and these are paired off to count the number of exchanged-pair contact-changes. Let n_1 be the number of remaining conformation 1 contacts after the maintained and the exchanged-partner contact-changes have been removed, and likewise n_2 be the number of remaining conformation 2 contacts. The number of exchanged-pair contact-changes was taken to be $N_{exchpair} = \min(n_1,n_2)$. In a DCG with maintained, exchanged-partner and exchanged-pair contact-changes removed there are only two-node subgraphs of one type left, either AB or BA. These represent the new contact-changes. The number of new contact-changes, N_{new}, is then given by $N_{new} = n_1 - N_{exchpair}$ or $N_{new} = n_2 - N_{exchpair}$, the former if $n_1 \geq n_2$, the latter if $n_2 > n_1$.

For the example in Figure 3B this process would result in $N_{maint} = 0$, $N_{exchpart} = 2$, $N_{exchpair} = 0$ and $N_{new} = 0$. For the less trivial case of DNA topoisomerase III shown in Figure 2, $N_{maint} = 7$, $N_{exchpart} = 13$, $N_{exchpair} = 7$ and $N_{new} = 4$.

Classifying domain movements

We classify domain movements according to which of the contact-change categories are non-empty or empty. There are five types of contact-change, but given that for all domain movements there are always residues that do not make interdomain contacts in both conformations, the no contact-change case is redundant. The only interesting case is when all residues are in this category but this case is covered when the number of contact-changes in all the other categories is zero. Therefore we need only consider the remaining four contact-change categories.

Each of the four categories can be empty or non-empty meaning there are sixteen (2^4) different classes. The no-contact class is when all four categories are empty. There are four "pure" classes, when only one category is non-empty, the other three being empty, e.g. "pure new" has $N_{maint} = 0$, $N_{exchpart} = 0$, $N_{exchpair} = 0$, $N_{new} \geq 1$. There are six classes when two categories are non-empty and two empty, e.g. "combined maintained, new" has $N_{maint} \geq 1$, $N_{exchpart} = 0$, $N_{exchpair} = 0$, $N_{new} \geq 1$. There are four classes when three categories are non-empty and one empty, e.g. "combined exchanged-pair, exchanged-partner, new" has $N_{maint} = 0$, $N_{exchpart} \geq 1$, $N_{exchpair} \geq 1$, $N_{new} \geq 1$. Finally, there is one class when all four categories are non-empty. These classes are given in Table 1 alongside the number of domain movements in each class.

It is interesting that there are so many examples of domain movements where no contacts are made between the domains (except at the hinge bending sites) in both conformations. Some of these may be due to domain linkers that act as rigid spacers between the domains to prevent unfavourable interdomain interactions during folding [19].

In terms of the total number of contact-change types across the whole set, there are 6810 new, 6087 maintained, 1448 exchanged-pair and 1150 exchanged-partner contact-changes.

Website for domain movement classification

We have produced a website where the domain movements are organised according to class (see http://www.cmp.uea.ac.uk/dyndom/class16). Each class comprises a list of protein names together with a pair of PDB accession codes and chain identifiers that specify the domain movement. The link provided takes one to a page where the DCG and a bar chart for the distribution of the number of instances in each of the four elemental contact-change categories are shown (see Figure 2). The number of independent regions is also given. The molecular graphics applet, Jmol (http://jmol.sourceforge.net/), is used to display the movement and to indicate the residues that make contact in each conformation. There is also a link to the corresponding DynDom page for that

Table 1. Numbers in each class.

Class	N° of examples
Pure no contacts	412
Pure maintained	56
Pure exchanged-partner	3
Pure exchanged-pair	9
Pure new	376
Combined maintained, exchanged-partner	10
Combined maintained, exchanged-pair	44
Combined maintained, new	225
Combined exchanged-partner, exchanged-pair	1
Combined exchanged-partner, new	34
Combined exchanged-pair, new	78
Combined maintained, exchanged-partner, exchanged-pair	35
Combined maintained, exchanged-partner, new	126
Combined maintained, exchanged-pair, new	137
Combined exchanged-partner, exchanged-pair, new	53
Combined all	223

doi:10.1371/journal.pone.0081224.t001

domain movement which gives details on the residues comprising the domains, the location of the hinge axis, the hinge-bending residues, the angle of rotation, percentage closure, as well as many other details, and a downloadable script for viewing the movement. A link to the DynDom family page is also provided which gives a conformational analysis of closely related structures and their domain movements [11].

Real domain movements and the model domain movements

In the Methods section we proposed an association between the elemental contact-changes and model domain movements. This association requires the domains and domain movements fulfil a set of conditions that are unlikely to be satisfied in real cases. Amongst others these conditions require the domains to be perfectly rigid and be convex in shape. It is clear from our results that many domain movements combine the four different types of elemental contact-changes suggesting immediately that the model domain movements are not appropriate for these cases. Even in the "pure" cases the model domain movements may not provide an appropriate description of the movement.

The model domain movement associated with the no contact set is the free domain movement implying the domains are free to move relative to each other but never make contact. This fact cannot be determined from just two structures and therefore we are unable to judge from our data whether the domains are free.

The pure new class implies the open-closed model movement; that is a movement that is predominantly a closure motion[17]. We can see an example that conforms to this model in Lysine-, Arginine-, Ornithine-binding (LAO) Protein (search for PDB accession codes 2LAO and 1LST on the main webpage). The protein has a well-defined hinge axis that brings the two rather globular domains together in a motion that is 99% closure. However, there are many examples in this class that do not conform to this model. An example can be seen in the domain movement in the human cellular receptor for Epstein-Barr virus (PDB codes 1GHQ and 1LY2) where contact is established via a twist motion (6.7% closure).

For the pure maintained class the corresponding domain movement is anchored and indeed only 12.5% of this class have rotations of more than 15° compared to 74.2% for the pure new indicating that maintained contacts do restrict rotation. However, because this group have small rotations domain demarcation becomes more subject to noise and many of these cases are due to only a slight difference in the rotational properties of the residues that maintain contact.

There are only three examples in the pure exchanged-partner class none of which are like the expected sliding twist model domain movement. The example of DnaA, a chromosomal replication initiator protein (PDB codes: 1L8Q and 2HCB), shows that an exchanged-partner contact-change can occur without a sliding twist movement if the interdomain screw axis is remote from the interdomain region, i.e. it violates one of the conditions for a model domain movement. A sliding twist movement is seen, however, in an immunoglobulin protein in the combined exchanged-partner, new class (PDB codes: 1E4K and 2IWG) where the domain movement is predominantly a twist (37% closure).

Finally in the pure exchanged-pair class which is associated with the see-saw model domain movement, six out of the nine examples would conform to a see-saw movement in that one can find a plane that the interdomain screw axis lies in and for which the contacts in the two conformations occur on either side of this plane. An example can be see for a histidine kinase (PDB codes: 1B3Q and 2CH4) which undergoes a clear see-saw movement with the domains rotating through 126°. An example that would not seem to be like a see-saw movement can be see for a lytic transglycosylase (PDB codes: 2G6G and 2G5D) where the non-globular shape of the domains and their location in relation to the hinge axis allows an exchanged-pair contact-change to occur via a non see-saw-like movement.

Discussion

We have used a contact analysis to help classify domain movements in proteins. The approach introduced here is based on identifying five types of elemental contact-changes. A real domain movement will comprise these elemental contact-changes but decomposing contact-changes in a real domain movement into the elemental contact-changes is non-trivial. A solution to this problem was found by encoding the contact-changes in a DCG and decomposing it in terms of the elemental DCGs which represent the elemental contact-changes. This allowed us to count the number of instances of each of the elemental contact-change types for each domain movement. This in turn has led to a classification system comprising sixteen classes.

Each elemental contact-change type can be related to a model domain movement. However, although some of those classified as "pure" in Table 1 may conform to a model domain movement most domain movements comprise a mixture of contact-change types and it is probably not correct to think of these as combining the model movements. The type of contact-change may be influenced by the size and flexibility of the residues, the local structure at the interdomain region, and its proximity to the hinge axis.

By counting disconnected subgraphs in a DCG, we are able to give the number of independent regions, that is, regions comprising sets of residues between which there are no contacts in either conformation. These regions may have a different role to play in the mechanism of the domain movement.

The elemental contact-change types may relate qualitatively to the energetics of domain movements. The no contact class suggests no energy need be expended in the movement (except perhaps in the hinge bending region). The "new" type suggests energy needs to be inserted into the system or is expended. A "maintained" type suggests a strong interaction with little or no energy consumed or expended or perhaps energy being consumed or expended to strain or relieve a maintained bond. An "exchanged-partner" type may suggest a low energy barrier if a sliding movement occurs because as one interaction is weakened the other is being strengthened. An "exchanged-pair" type by contrast may indicate an energy barrier if one interaction is broken before the other one is formed in a see-saw movement. However, many domain movements are highly complex and this kind of simple interpretation will obviously not always apply. Indeed, one can imagine the exchanged-pair contact-change occurring in a way much like the sliding case if as one pair of contacts is being lost another pair of contacts is being gained such that there is no appreciable energy barrier. The work by Sinha et al. [16] suggests this mechanism with the finding that for a number of domain proteins the nonpolar buried surface area in the open state matches or slightly exceeds the nonpolar buried surface area in the closed state, especially when the domain movement is small.

For enzymes it has been shown that the type of structural change can relate to the type of reaction being catalysed [7] and it will be of interest to determine the relationship between the type of domain movement according to the classification scheme used here and molecular function.

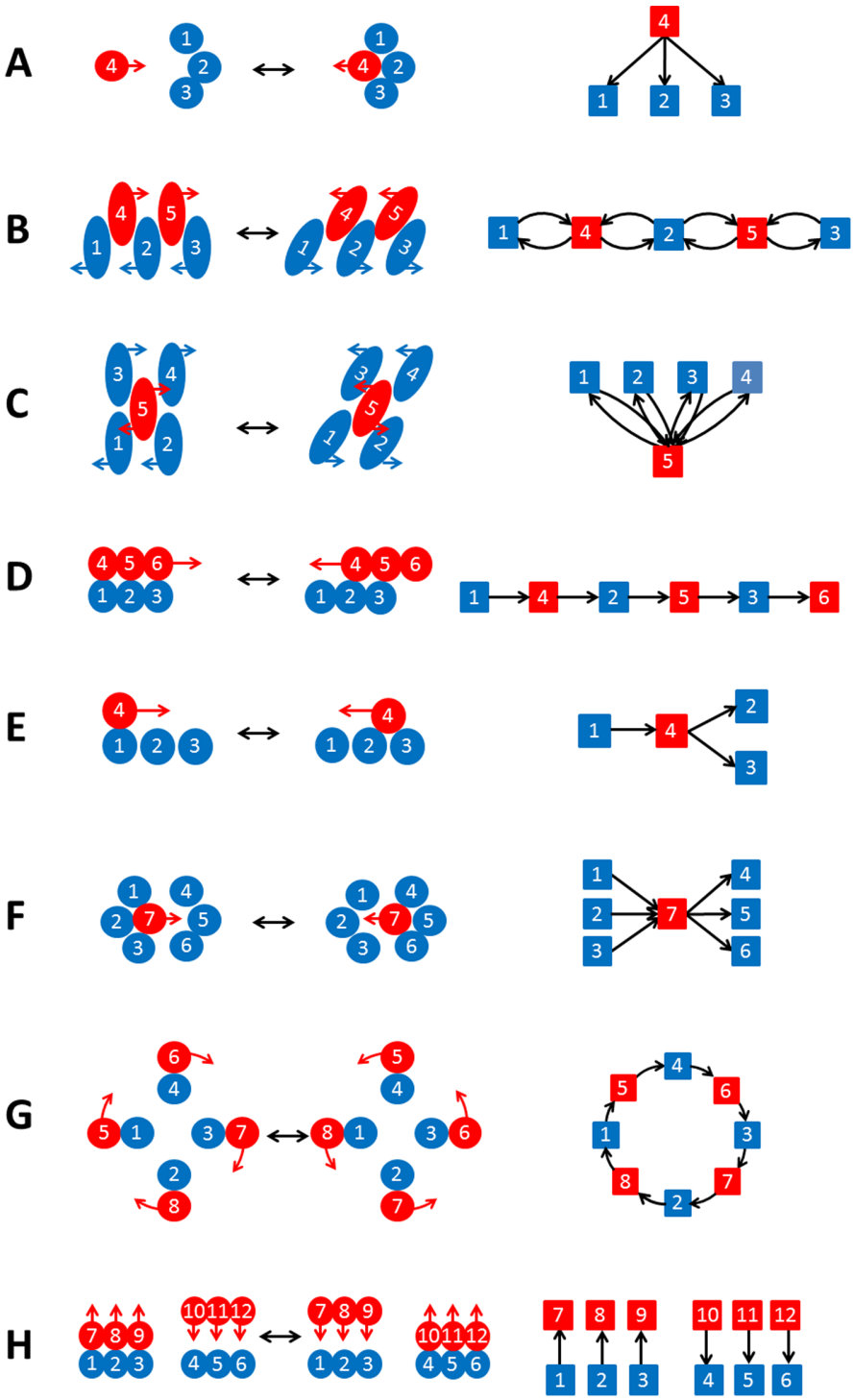

Figure 4. Motifs in DCG's indicating possible mechanism. Each filled circle or ellipse indicates a residue with domain A residues coloured blue and domain B residue red. Touching circles or ellipses indicate a contact. The graphs with squares and arrows are the associated DCGs. (A) "Multiple new." A residue moves from having no contacts in one conformation to having multiple contacts in the other conformation. (B) "Linear Interlocking." This might occur when there is a "shear" movement according to Gerstein et al. [4]. The interlocking side chains are depicted in a sequence of doubly linked nodes in the DCG suggesting strong bonds that cannot be broken. (C) "Anchoring residue." Here a single residue maintains contact with a number of other residues from the other domain, acting possibly as an anchor. (D) "Linear slide." Here residues slide relative to each other each making at most one contact in both conformations. The DCG depicts a set of singly connected nodes arranged linearly. (E) "Branched slide." Here one residue makes a single contact in one conformation but two contacts in the other giving a branched DCG. (F) "Multiple-to-Multiple slide." A residue moves from having multiple contacts with a set of residues in one conformation to multiple contacts with another set of residues in the other conformation. The DCG is clearly suggestive of this process. (G) "Closed-cycle slide." If the domains have a twisting movement as depicted on the left the DCG will have a closed cycle. (H) "Multiple see-saw." A see-saw movement as depicted on the left will have a DCG with edge-arrows that clearly suggest a see-saw movement.
doi:10.1371/journal.pone.0081224.g004

Although we have used the DCGs to classify domain movements, they should provide, in themselves, a great deal of insight in individual cases, especially when considered by experts on the protein concerned. In essence they give a visual metaphor for the movement and its mechanism. Here we consider motifs that appear in DCGs indicating particular mechanisms.

Multiple new: A residue with no contact in one conformation moves into a pocket making multiple contacts in the other conformation. The associated graph is shown in Figure 4A and is a clearly recognisable motif. The domain movement in aclacinomycin 10-hydroxylase provides an example (structural pair: 1XDS, chain A; 1QZZ, chain A).

Linear Interlocking: A sequence of interlocking residues as depicted in Figure 1 of reference 4 for a shear movement would have a graph as shown in Figure 4B with a series of doubly linked nodes. The doubly linked nodes, give the visual metaphor of strong contacts between residues that cannot be broken. This motif is easily seen in a visual scan of a DCG. Tryptophanyl-tRNA synthetase (structural pair: 1MAU, chain A; 1I6M, chain A) provides an example.

Anchoring residue: A single residue maintains contact with a number of other residues during the domain movement, acting perhaps as an anchor as shown in Figure 4C. The domain movement in glucokinase provides an example (structural pair: 1Q18, chain A; 1SZ2, chain B).

Linear slide: A region from domain B (red in Figure 4D) sliding on a region from domain A (blue) has a graph with a series of singly linked nodes with edges all pointing in the same direction. One can think of the region from domain B sliding on the surface provided by the region of domain A with the direction of the edges indicating the direction of the movement of domain B in going from conformation 1 to conformation 2, e.g. residue 4 is moving from residue 1 to residue 2. Again the graph gives a visual metaphor for a simple sliding movement and is an easily recognised motif. The domain movement in human IGG1 FC fragment provides an example (structural pair; 1E4K, chain B; 1IWG, chain A).

Branched slide: If a residue in domain B makes a single contact with a residue in domain A in conformation 1 but makes contact with two residues in domain A in conformation 2 then the graph will have a branch as shown in Figure 4E. The movement in a MHC class I molecule provides an example (structural pair: 1ZT7, chain C; 1MWA, chain I).

Multiple-to-multiple slide: If in conformation 1 a residue in domain B makes multiple contacts with residues in domain A and moves to make multiple contacts with another region of domain A in conformation 2, the graph will be like that shown in Figure 4F. Again the graph provides a clear visual metaphor of the type of contact-change that occurs. NADH pyrophosphatase provides an example (structural pair: 1VK6, chain A; 2GB5, chain A).

Closed-cycle slide: If the two domains undergo a rotational motion, such that the two surfaces remain in contact, i.e. a twisting motion, and individual residues undergo a sliding movement where *every* residue makes a single contact in both conformations, then the graph will be a closed cycle as shown in Figure 4G. The associated graph clearly indicates such a rotational motion, providing a visual metaphor for the movement and an easily recognisable motif. There is always an even number of residues involved in this motif. The photosynthetic reaction centre from Thermochromatium tepidum provides an example (structural pair: 2EYT, chain A; 2EYS, chain A). As one might expect the movement in this protein is predominantly a twist (33.5% closure).

Multiple see-saw: If a region makes contact in conformation 1 but not in conformation 2, and a completely separate region, makes contact in conformation 2 but not in conformation 1, then the graph will look like that shown in Figure 4H. This will occur when the domains undergo a see-saw motion. The associated graph provides a strong visual metaphor for a see-saw movement. The domain movement in maltodextrin binding protein provides an example (structural pair: 1MDP, chain 2; 2OBG, chain A).

Our approach considers contacts between residues within the same subunit even if the protein functions as a multimer. Although our understanding is that domain movements in multimeric proteins involve more intrasubunit contact-changes than intersubunit contact-changes, intersubunit contact-changes need to be included in the future. The current approach was necessitated by the use of the NRDPDM which was constructed using DynDom which is only able to analyse domain movements in individual subunits. The use of a new program, DynDom3D [20], designed to analyse domain movements in multimers, will remedy this. A related issue is the absence of residue-ligand contacts in the DCGs when the ligand concerned induces the domain closure. From the viewpoint of the energetics of domain closure, the inclusion of residue-ligand contacts in the DCG would be essential, but when DCGs are used for the purpose of classifying the domain movements (e.g. whether a see-saw or a sliding-twist movement) the inclusion of these contacts should not be necessary.

Although we have limited our study to experimentally determined structures, these methods could be applied to the results of Molecular Dynamics (MD) simulation and Normal Mode Analysis (NMA). In the case of NMA a single normal mode eigenvector can be represented by two structures from which residue contacts or perhaps energy-based thresholds could be used to define the DCG. Likewise in the case of MD simulation principal component analysis gives eigenvectors from which two extreme structures can be created.

DCGs provide us with a way to identify motifs related to movements of domains. However, DCGs need not be confined to the analysis of domain movements but can be applied to any case where there are two conformations and two sets of objects e.g. subunits that have different associations in the two conformations.

Acknowledgments

We thank Russell Smith for help with constructing the website.

Author Contributions

Conceived and designed the experiments: SH. Performed the experiments: DT SH GC. Analyzed the data: DT SH GC. Contributed reagents/materials/analysis tools: DT SH GC. Wrote the paper: SH. Conceived approach and method development: SH. Primary data analysis and method development: DT SH. Method development: GC.

References

1. Bennet WS, Huber R (1984) Structural and functional aspects of domain motions in proteins. Critical Review in Biochemistry 15: 291–384.
2. Schulz GE (1991) Domain motions in proteins. Current Opinion in Structural Biology 1: 883–888.
3. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, et al. (2000) The Protein Data Bank. Nucleic Acids Research 28: 235–242.
4. Gerstein M, Lesk AM, Chothia C (1994) Structural mechanisms for domain movements in proteins. Biochemistry 33: 6739–6749.
5. Gerstein M, Krebs W (1998) A database of macromolecular motions. Nucleic Acids Research 26: 4280–4290.
6. Gutteridge A, Thornton J (2005) Conformational changes observed in enzyme crystal structures upon substrate binding. Journal of Molecular Biology 346: 21–28.

7. Koike R, Amemiya T, Ota M, Kidera A (2008) Protein structural change upon ligand binding correlates with enzymatic reaction mechanism. Journal of Molecular Biology 379: 397–401.
8. Hayward S, Berendsen HJC (1998) Systematic analysis of domain motions in proteins from conformational change: New results on citrate synthase and T4 lysozyme. Proteins 30: 144–154.
9. Hayward S, Lee RA (2002) Improvements in the analysis of domain motions in proteins from conformational change: DynDom version 1.50. Journal of Molecular Graphics and Modelling 21: 181–183.
10. Hayward S (1999) Structural principles governing domain motions in proteins. Proteins 36: 425–435.
11. Qi G, Lee RA, Hayward S (2005) A comprehensive and non-redundant database of protein domain movements. Bioinformatics 21: 2832–2838.
12. Qi GY, Hayward S (2009) Database of ligand-induced domain movements in enzymes. BMC Structural Biology 9.
13. Amemiya T, Koike R, Fuchigami S, Ikeguchi M, Kidera A (2011) Classification and Annotation of the Relationship between Protein Structural Change and Ligand Binding. Journal of Molecular Biology 408: 568–584.
14. Amemiya T, Koike R, Kidera A, Ota M (2012) PSCDB: a database for protein structural change upon ligand binding. Nucleic Acids Research 40: D554–D558.
15. Brylinski M, Skolnick J (2008) What is the relationship between the global structures of apo and holo proteins? Proteins-Structure Function and Bioinformatics 70: 363–377.
16. Sinha N, Kumar S, Nussinov R (2001) Interdomain interactions in hinge-bending transitions. Structure 9: 1165–1181.
17. Hayward S, Kitao A, Berendsen HJC (1997) Model free methods to analyze domain motions in proteins from simulation. A comparison of a normal mode analysis and a molecular dynamics simulation of lysozyme. Proteins 27: 425–437.
18. Gerstein M, Lesk AM, Baker EN, Anderson B, Norris G, et al. (1993) Domain closure in lactoferrin: Two hinges produce a see-saw motion between alternative close-packed interfaces. Journal of Molecular Biology 234: 357–372.
19. George RA, Heringa J (2002) An analysis of protein domain linkers: their classification and role in protein folding. Protein Engineering 15: 871–879.
20. Poornam GP, Matsumoto A, Ishida H, Hayward S (2009) A method for the analysis of domain movements in large biomolecular complexes. Proteins 76: 201–212.

Bioinformatics Advance Access published August 14, 2014

BIOINFORMATICS ORIGINAL PAPER

2014, pages 1–8
doi:10.1093/bioinformatics/btu506

Structural bioinformatics

Advance Access publication July 30, 2014

Quantitative method for the assignment of hinge and shear mechanism in protein domain movements

Daniel Taylor, Gavin Cawley and Steven Hayward*

D'Arcy Thompson Centre for Computational Biology, School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: A popular method for classification of protein domain movements apportions them into two main types: those with a 'hinge' mechanism and those with a 'shear' mechanism. The intuitive assignment of domain movements to these classes has limited the number of domain movements that can be classified in this way. Furthermore, whether intended or not, the term 'shear' is often interpreted to mean a relative translation of the domains.

Results: Numbers of occurrences of four different types of residue contact changes between domains were optimally combined by logistic regression using the training set of domain movements intuitively classified as hinge and shear to produce a predictor for hinge and shear. This predictor was applied to give a 10-fold increase in the number of examples over the number previously available with a high degree of precision. It is shown that overall a relative translation of domains is rare, and that there is no difference between hinge and shear mechanisms in this respect. However, the shear set contains significantly more examples of domains having a relative twisting movement than the hinge set. The angle of rotation is also shown to be a good discriminator between the two mechanisms.

Availability and implementation: Results are free to browse at http://www.cmp.uea.ac.uk/dyndom/interface/.

Contact: sjh@cmp.uea.ac.uk.

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on January 31, 2014; revised on July 8, 2014; accepted on July 18, 2014

1 INTRODUCTION

Multi-domain proteins can be regarded as comprising quasi-globular regions connected by linkers that allow their relative movement. Consequently, domain movements are often engaged in protein function in a wide variety of contexts, including catalysis, transport, signaling and immune response (Bennet and Huber, 1984; Gerstein *et al.*, 1994; Schulz, 1991). In many of these cases, domain movements occur on the binding of a ligand. For example, in multi-domain enzymes, the binding of the substrate in the interdomain cleft causes the domains to close trapping the substrate in the specific environment necessary for catalysis. Well-known examples include citrate synthase (Wiegand and Remington, 1986), liver alcohol dehydrogenase (Eklund *et al.*, 1981) and F1-ATPase β subunit (Abrahams *et al.*, 1994).

Experimentally determined information on protein domain movements at the atomic level comes from the structures of proteins in different states solved primarily by X-ray crystallography and nuclear magnetic resonance spectroscopy. These different states may relate to function when they are within the functional cycle, but they may also be due to differences in the experimental conditions under which the structures were solved, or could be due to natural or engineered mutations. These structures, deposited in the Protein Data Bank (PDB) (Berman *et al.*, 2000), are a rich source of information on protein domain movements. Thus, multiple structures of proteins have been used to analyse and classify domain movements in a number of studies over the past 20 years (Amemiya *et al.*, 2011; Brylinski and Skolnick, 2008; Gerstein *et al.*, 1994; Hayward, 1999; Qi and Hayward, 2009; Sinha *et al.*, 2001; Taylor *et al.*, 2013).

The concepts of hinge and shear mechanisms in domain movements were first described by Gerstein *et al.* (1994) in their influential review article. Subsequently, the DataBase of Macromolecular Movements (DBMM) appeared online with further examples (Gerstein and Krebs, 1998). Hinge motions were described as those where the domains approach each other perpendicular to the plane of the interface. Shear movements, in contrast, have a preserved domain interface where the domains have a relative movement along the plane of the interface. Hinge movements would allow for large relative movement of the domains, whereas shear movements would be limited by the preserved side-chain packing at the interface. Although few details were given, it seems that these assignments were made intuitively, probably using molecular graphics software to compare the open and closed structures. This approach obviously limits the number of cases that can be classified in this way, and is also open to criticism in that it is not reproducible. Despite these limitations, the fact remains that, for some proteins, domain closure occurs through a simple 'pacman' opening-closing movement, whereas for others the movement is more complex with the two domains remaining in contact during the domain movement. To investigate this further, one would need to develop an automatic method for assigning hinge and shear that uses quantitative and reproducible methods. With this method, one would be able to classify a much larger number of domain movements allowing the further investigation of these two types of mechanisms. To do this, quantities are required that capture the essential difference between hinge and shear movements. The descriptions used in the articles that

*To whom correspondence should be addressed.

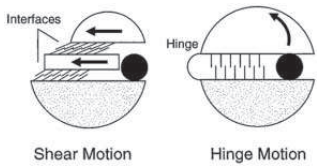

Fig. 1. Shear and hinge mechanisms. Based on the depiction given in Figure 1 in Gerstein *et al.* (1994) illustrating the shear and hinge mechanisms. The arrows indicate the direction of movement from the closed (depicted) to the open conformation

describe the hinge and shear movements point to two alternative approaches: one based on the relationship between the domain interface and the movement, the other based on residue contact changes (e.g. via 'interdigitating sidechains', or newly established contacts, see Fig. 1). In this article, we have taken the latter approach.

In our previous work (Taylor *et al.*, 2013), changes in inter-domain residue contacts that occur in the domain movement were used to define four types of elemental contact changes: maintained, exchanged partner, exchanged pair and new. A maintained contact change is where the same pair of residues is found to be in contact in both conformations. An exchanged-partner contact change is one where the same residue is found to be in contact with two different residues in the two conformations, as would occur in a sliding movement. An exchanged-pair contact change is one where the residue contact pair in one conformation and the residue contact pair in the other conformation have no residues in common, as would occur in a see-saw movement. A new contact change is one where there is a contact pair in one conformation but no contact pair in the other conformation and might occur in an open to closed domain movement. Counting the number of instances of each elemental contact-change type is non-trivial, but a solution was found by the use of so-called 'dynamic contact graphs' (Taylor *et al.*, 2013). If a domain movement is predominantly shear, one would expect it to have a relatively large number of either maintained or exchanged-partner contact changes, whereas if a domain movement is predominantly hinge, then one would expect it to have a relatively large number of exchanged-pair or new contact changes.

Here machine learning is used, which uses the number of instances of each of these four types of contact changes for each domain movement to 'learn' from the DBMM to make hinge and shear assignments optimally. The movements in a much larger dataset can then be assigned to hinge and shear categories automatically. In a sense, this approach has allowed us to extract some essence of the subjective approach used to assign hinge and shear movements in the DBMM so that these assignments can be made to a larger dataset.

The language, and the figure used in the review article by Gerstein *et al.* (1994) to depict the shear movement, appears to have led to an interpretation of a shear movement to mean a relative translational movement of the domains, i.e. there is little or no rotational movement involved. Figure 1 illustrates hinge and shear movements based on the figure and descriptions given in the review article (Gerstein *et al.*, 1994). A similar figure has appeared in a review article on protein flexibility and drug design (Teague, 2003).

One might wonder why it is important to make a distinction between a rotational motion and a translational motion in the context of protein domain motions. The key point is that rotations will be locally controlled at specific hinge sites, whereas a translational motion would not be controlled at specific sites. Sites where control over a functional movement is exercised are potential target sites for therapeutic molecules. For example, a drug molecule binding to a single hinge site in an enzyme might prevent domain closure and subsequent catalysis of the natural substrate occurring just as effectively as an inhibitor that binds to the active site. The assignment of a domain movement as occurring via a translation would seem to preclude it from this form of alternative drug-site targeting.

2 MATERIALS AND METHODS

The basic data are the 2035 unique domain movements from the non-redundant database of protein domain movements, NRDPDM (Qi *et al.*, 2005). The domain movements were determined by the DynDom program (Hayward and Berendsen, 1998; Hayward and Lee, 2002). These unique movements come from 1578 families, which means that some domain movements are from the same family. Individual cases from this dataset are available to browse at http://www.cmp.uea.ac.uk/dyndom. To simplify the analysis, only those cases with two domains were used. Of the 2035 cases, 1822 are two-domain proteins. This dataset will be referred to as 'NRDPDM2d'.

DBMM (Gerstein and Krebs, 1998) is available online (http://www.molmovdb.org) and has 37 examples of domain motions classified as 'predominantly shear' and 75 examples of domain motions classified as 'predominantly hinge'.

2.1 Residue contact definition

Contact between residue i and residue j means any heavy atom of residue i is within 4 Å of any heavy atom of residue j. However, before the set of pair-wise contacts between residues in each domain and for each conformation is determined, residues at the boundaries of the domains assigned by DynDom as bending regions were removed, as were residues close to the interdomain screw axis (any heavy atom of the residue within 5.5 Å of the axis). The reason for this is that they would be expected to have maintained contacts irrespective of the nature of the domain movement.

2.2 Counting the number of elemental contact changes in a domain movement

Let $\{(a_{1i},b_{1i})\}$, $i = 1, N_1$ be the set of ordered pairs of residue numbers corresponding to residues, a_{1i} from domain A, and b_{1i} from domain B, making a contact in conformation 1. Let $\{(a_{2i}, b_{2i})\}$, $i = 1, N_2$ be the equivalent set for conformation 2. From these two sets, a 'dynamic contact graph' (DCG) can be created as described by Taylor *et al.* (2013). A DCG is a directed graph, an example of which from citrate synthase is shown in Figure 2A. In a DCG, each node of the graph represents a residue of which there are two types: those in domain A and those in domain B. An edge joins the two nodes when there is a contact between the residue in domain A and the residue in domain B, with the edge direction being from the node in A to the node in B if a contact exists in conformation 1 ($a_{1i} \rightarrow b_{1i}$) and in the opposite direction if the contact exists in conformation 2 ($a_{2i} \leftarrow b_{2i}$). Figure 2B shows the 'elemental DCGs' and the elemental contact changes they represent, namely, maintained, exchanged-partner, exchanged-pair and new. As outlined by

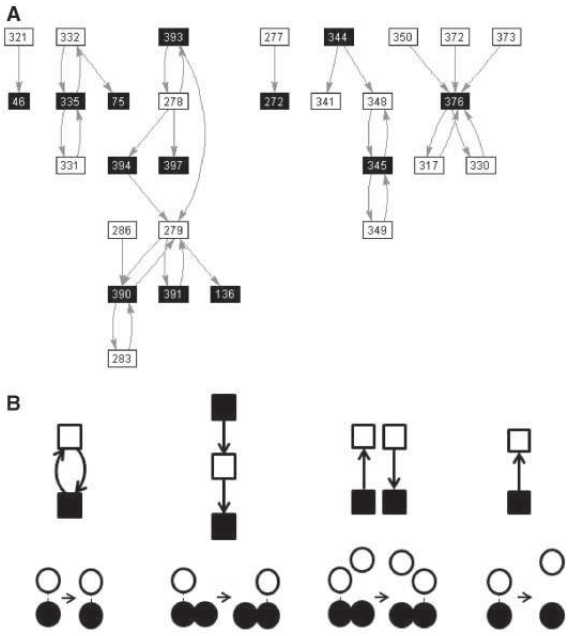

Fig. 2. DCG and decomposition. (**A**) The DCG for the domain movement between conformation 1 (PDB accession code: 1CTS) and conformation 2 (PDB accession code: 1CSH) in citrate synthase. A filled square corresponds to a residue in domain A, and an open square corresponds to a residue in domain B with the residue number written in the square. An arrow from a residue in A to a residue in B indicates a contact between the residues in conformation 1. An arrow from a residue in B to a residue in A indicates a contact between the residues in conformation 2. (**B**) The elemental DCGs for 'maintained', 'exchanged-partner', 'exchanged-pair' and 'new' that represent the pairwise residue contact changes depicted underneath each graph. The DCG in (A) is decomposed into these elemental DCGs to give $N_{maint} = 10$, $N_{exchpart} = 2$, $N_{exchpair} = 2$ and $N_{new} = 6$. The prediction value for this domain movement is 0.55, which puts it in the Mixed class

Taylor *et al.* (2013), any complex DCG can be decomposed into these elemental DCGs, which allows us to count the number of elemental contact changes involved in the movement. The number of elemental contact changes, N_{maint}, $N_{exchpart}$, $N_{exchpair}$ and N_{new} [referred to collectively as **N** where $\mathbf{N} = (N_{maint}\ N_{exchpart}\ N_{exchpair}\ N_{new})$], is the primary input for the logistic regression.

2.3 Logistic regression

Matching domain pairs between DBMM and NRDPDM2d To perform logistic regression, pairs of structures representing the domain movement in NRDPDM2d need to be matched to pairs of structures in DBMM. NRDPDM is organized by protein family within which the structures are grouped according to a conformational clustering procedure (Qi *et al.*, 2005). We considered there to be a match between a pair of structures in NRDPDM2d and DBMM if both DBMM structures (identified by PDB accession code and chain identifier) are found in the same NRDPDM family.

Logistic regression procedure Let $\mathbf{N}^i$ represent a four-component vector with $N^i_1 = N_{maint,i}$, $N^i_2 = N_{exchpart,i}$, $N^i_3 = N_{exchpair,i}$ and $N^i_4 = N_{new,i}$, where $N_{maint,i}$, $N_{exchpart,i}$, $N_{exchpair,i}$ and $N_{new,i}$, denote N_{maint}, $N_{exchpart}$, $N_{exchpair}$ and N_{new} in domain movement i, respectively. Let $t^i = 0$ when the DBMM assignment for domain movement i is predominantly hinge, and $t^i = 1$ when the DBMM assignment for domain movement i is predominantly shear. Given labelled training data $D = \{(\mathbf{N}^i, t^i)\}$, logistic regression constructs a decision rule that can be used to distinguish between objects belonging to two classes. The logistic regression model is of the form:

$$\mathrm{logit}(y(\mathbf{N})) = \mathbf{w} \cdot \mathbf{x} + b \tag{1a}$$

where

$$\mathrm{logit}(p) = \log\left(\frac{p}{1-p}\right) \tag{1b}$$

w is a four-component vector of regression coefficients and b is a scalar bias parameter. The optimal value of the regression coefficients is determined by minimizing the cross-entropy training criterion:

$$E = -\frac{1}{2}\sum_{i=1}^{L}[t^i \log(y^i) + (1-t^i)\log(1-y^i)] \tag{2}$$

where $y^i = y(\mathbf{N}^i)$, L is the total number of domain movements in the training set (i.e. the total number of NRDPDM2d domain movements corresponding to the DBMM set).

The output of the logistic regression model can then be regarded as an estimate of the Bayesian a posteriori probability of class membership, i.e.

$$y(\mathbf{N}) \approx P(t = 1 \mid \mathbf{N}) \tag{3}$$

2.4 Translation and Chasles' theorem

Chasles' theorem (Chasles, 1830) states that the most general displacement of a rigid body is a screw movement about a unique screw axis. That is, given a rigid body in two different positions (and orientations), the body can be taken from one to the other by a screw movement about a unique screw axis. The DynDom program (Hayward and Berendsen, 1998) determines this screw axis. DynDom produces a PDB-formatted file that contains the structures superposed on one domain together with an 'arrow molecule' that depicts the interdomain screw axis. This file allows the calculation of distances between the structures and the interdomain screw axis and can be used for visualizing the domain movement using molecular graphics software. DynDom also gives the rotation angle and translational displacement along the axis that occurs in the screw movement. If the movement is a pure rotation about an axis, then this screw axis is the rotation axis. If a body undergoes a rotation about a structural hinge but also undergoes a translation in the plane of the rotation, then the interdomain screw axis will not coincide with the original hinge axis. Thus, we test for the screw axis being located outside the body of the protein. If this is the case, then we can be sure that there is no control over the rotation being exercised at the axis location, and consequently any rotation about a structural hinge must be accompanied by a translation in the rotation plane. The location of the interdomain screw axis was previously used to define a 'mechanical hinge' (Hayward, 1999), it being a bending region (a region of the backbone connecting the two domains within which the rotational transition occurs) with any one of its C^α-atoms within 5.5 Å of the interdomain screw axis. In proteins not all bending regions are mechanical hinges, but those that are can be thought of as controlling the domain movement much as the hinge of a door helps to determine the location of its rotational axis. An interdomain screw axis that has at least one mechanical hinge has been called an 'effective hinge axis' (Hayward, 1999). DynDom also determines the percentage closure. Those with a percentage >50% are annotated here as having a closure motion; those with a percentage ≤50% are annotated as having a twisting motion.

The significance tests made are described in the Supplementary Material.

3 RESULTS

3.1 Prediction of hinge and shear

Of the 37 'predominantly shear' domain movements in the DBMM, 21 were also in NRDPDM2d, and of the 75 'predominantly hinge' domain movements in the DBMM, 41 were also in NRDPDM2d. To improve statistics, we used the DynDom program directly on structures provided at the DBMM, which gave an extra two examples to add to the 21 from NRDPDM2d in the shear category and an extra 13 to add to the 41 in NRDPDM2d in the hinge category. The training set can be found in the Supplementary Material. The $\mathbf{N}^i$ were calculated for each of the 77 domain movements in the training set, and logistic regression was carried out as described in the Methods section. Logistic regression produced the following model:

$$y(\mathbf{N}) = \frac{1}{1+e^{\alpha}} \tag{4a}$$

where

$$\alpha = -0.2387N_{maint} - 0.0356N_{exchpart} + 0.4249N_{exchpair} + 0.2122N_{new} + 0.1467 \tag{4b}$$

To determine whether this model corresponds well to the DBMM assignments, a receiver-operating characteristic curve (ROC) curve was determined. A ROC curve plots the true-positive rate against the false-positive rate. A true positive is a shear correctly predicted shear, and a false positive is a hinge incorrectly predicted shear. The true-positive rate is the number of true positives to number of shear in the dataset, and the false-positive rate is the number of false positives to number of hinge in the dataset. Figure 3A shows the ROC curve. The area under the ROC curve is 0.83, indicating that the logistic function is a good discriminator between hinge and shear movements. To confirm this result, a leave-one-out cross-validation approach was used, the ROC curve of which is shown in Figure 3B. The area under this ROC curve is 0.77, confirming that the logistic function is able to give a good predictor for hinge and shear. Regularized logistic regression (Cessie and Houwelingen, 1992) and kernel logistic regression (Cawley *et al.*, 2007; Cawley and Talbot, 2008) were also tried, but these did not improve on the results obtained using conventional logistic regression.

Before Equation 4 was applied to the NRDPDM2d, the 412 cases where $\mathbf{N} = \mathbf{0}$, were removed, i.e. those cases where N_{maint}, $N_{exchpart}$, $N_{exchpair}$ and N_{new} are all equal to zero. The removed movements are those classified as 'No-contact', as there are no domain contacts in either conformation. These cases would not be expected to be classed as either shear or hinge according to Gerstein *et al.*, and no such case was found among the 77 DBMM examples. Equation 4 was applied to the remaining 1410 movements in the NRDPDM2d.

Figure 4A shows a histogram for the frequency distribution of the prediction values y. As can be seen, there is no obvious clustering, but there are pronounced peaks at certain values of y. The peaks labelled a,b,c,d,e are due to domain movements where $\mathbf{N} = (0\ 0\ 0\ N_{new})$, $N_{new} = 1,2,3,4,5$, respectively. In our previous work (Taylor *et al.*, 2013), these domain movements are in the 'Pure new' class (the most populous after the 'No-contact' class),

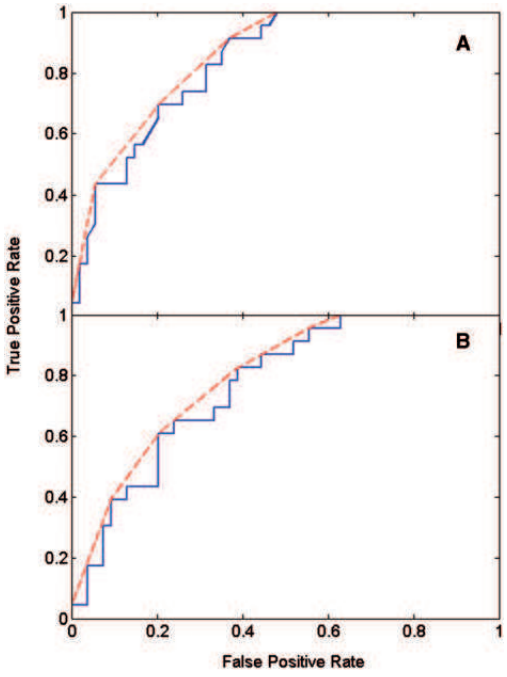

Fig. 3. ROC curves for the prediction of hinge and shear using logistic regression. A predictor for shear and hinge was constructed and tested against predominantly shear and predominantly hinge assignments in the DBMM. The ROC curve for the logistic function, given in Equation 4, gives the unbroken line; the convex hull of the unbroken line is the broken line. (A) The area under the ROC curve is 0.83, and the area under the convex hull is 0.86. (B) The ROC curve for a leave-one-out cross-validation approach. The area under the ROC curve is 0.77, and the area under the convex hull, 0.80

meaning that in one conformation there are no contacts between the domains and in the other conformation there are exactly N_{new} pairwise residue contacts. For these cases, the larger the N_{new}, the more 'hinge-like' they seem to become in terms of their y value (decreasing with increasing N_{new}), although arguments based on the presence or absence of contacts alone might conclude they are all equally domain movements via a hinge mechanism; for all of these, $y < 0.45$. The peak f, at $y = 0.470$, is due to the predominance of examples with $\mathbf{N} = (1\ 0\ 0\ 1)$, which are from the 'Combined maintained new' class (the third most populous class). The peak g, at $y = 0.523$, is from the 'Pure maintained' class with $\mathbf{N} = (1\ 0\ 0\ 0)$ where only one pairwise residue contact is maintained between the domains in the domain movement.

Given that we would like to include all cases in the 'Pure new' class as examples of a domain movement via a hinge mechanism, but to be sure that we are excluding weak examples from our classifier, the domain movements were put into three classes as follows:

> 'Hinge', for cases with $0 \leq y \leq 0.45$; 'Shear', for cases with $0.55 \leq y \leq 1.0$; 'Mixed', for cases with $0.45 < y < 0.55$.

It is important for the comparisons we intend to make that the two main classes, hinge and shear, have a high precision.

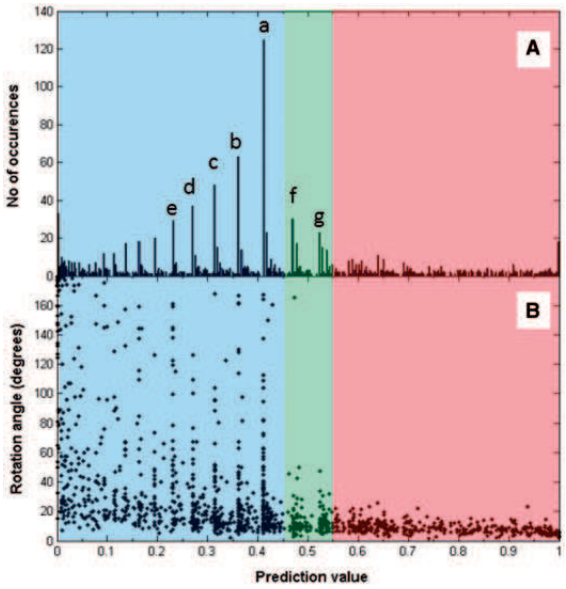

Fig. 4. Prediction value distributions. 'Hinge', 'Mixed' and 'Shear' are in the prediction value regions 0.0–0.45, 0.45–0.55 and 0.55–1.0, respectively. (**A**) Histogram of prediction values. The spikes indicated by 'a', 'b', 'c', 'd', 'e', 'f' and 'g' correspond to **N** = (0 0 0 1), **N** = (0 0 0 2), **N** = (0 0 0 3), **N** = (0 0 0 4), **N** = (0 0 0 5), **N** = (1 0 0 1) and **N** = (1 0 0 0), respectively. (**B**) The rotation angle plotted against prediction value. The same peaks can be seen and offer an explanation for their existence. For example, the peak at 'a' for prediction value 0.411 corresponding to **N** = (0 0 0 1) means there are a large number of domain movements with various angles of rotation that are all able to break a single residue contact pair

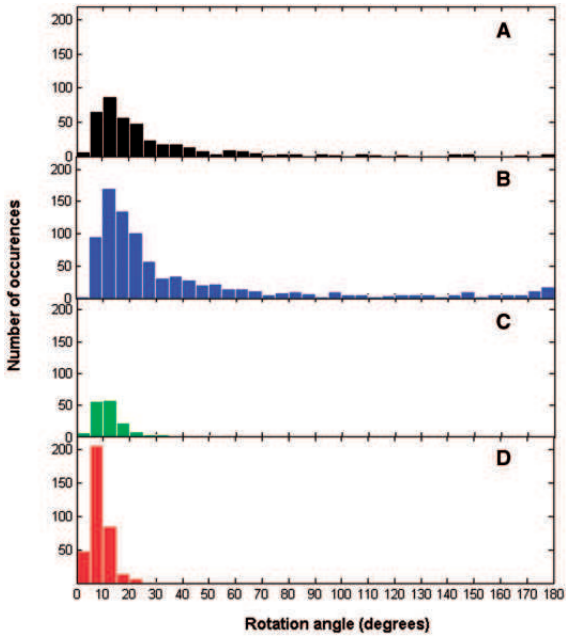

Fig. 5. Histograms for rotation angles. (**A**) no-contact set, (**B**) hinge set, (**C**) mixed set, (**D**) shear set

The precision of a class can be calculated as the proportion of cases *correctly* predicted to be in that class (true-positive results) to the total number cases predicted to be in that class. Of the 61 DBMM cases predicted hinge, 48 were actually predominantly hinge according to DBMM, giving a precision of 79%. The numbers are low for the calculation of the precision of shear prediction. Only 12 DBMM cases were predicted shear, with 9 of them actually classed as predominantly shear by DBMM, giving a precision of 75%. The natural boundary of 0.5 (so hinge for $0 \leq y \leq 0.5$ and shear for $0.5 < y \leq 1.0$) lowers the precision for the shear class to below 70%. These results support our choice of 0.45 and 0.55 as the classification boundaries and show that we are able to assign hinge and shear to domain movements automatically with a high degree of correspondence with assignments made using the intuitive method.

Applying the predictor to the 1410 examples, 884 are the hinge class (63%), 361 in the shear class (26%), with the remaining 165 in the mixed class (12%). Out of the whole set of 1822 domain movements, 23% are in the No-contact set, 49% hinge, 20% for shear, and 9% mixed. This means we have a 10-fold increase in the number of examples over the number previously available allowing us to study hinge and shear mechanisms using statistical methods to measure the significance of our results. The result of applying the predicator to the training set can be found in the Supplementary Material.

3.2 Rotation angle as indicator of hinge and shear

Figure 4B shows the rotation angle plotted against the prediction value. One can discern a general trend for the rotation angle to increase with decreasing prediction value, i.e. the motions become more hinge-like. Large rotations occur below a prediction value of 0.45 in the hinge region. Most of the peaks there correspond to the peaks indicated in Figure 4A and also correspond to the 'Pure new' class. In fact, nearly 80% of those peaks in hinge are where N_{new} is larger than N_{maint}, $N_{exchpart}$, and $N_{exchpair}$. Figure 5 shows histograms for the rotation angles for the four categories. One can immediately see that for shear, rotations do not exceed 25°. For these cases, there is nearly always either predominance in the number of maintained, N_{maint}, or the number of exchanged-partner contact changes, $N_{exchpart}$, indicating that for a preserved-interface movement the angle of rotation is limited to 25°.

Also of interest in Figure 5 is the slight increase in the number of hinge examples where the angle of rotation is close to 180°. Some of these are examples of domain swapping (Bennett *et al.*, 1994).

Figures 4 and 5 suggest that the angle of rotation is predictive of whether a domain movement is hinge or shear. Figure 6 shows the extent to which rotation can be used for predicting hinge or shear. In Figure 6A, the blue line gives, among all domain movements (excluding non-contact cases) with rotation angles greater than or equal to any selected threshold value, the proportion that are from the hinge class. It shows that among the set of domain movements (excluding non-contact cases) with rotation angles $\geq 10°$, 80% are hinge. In Figure 6B, the red line gives, among all domain movements (excluding non-contact cases) with rotation angles less than any selected threshold value, the proportion that are from the shear class. It shows that among the set of

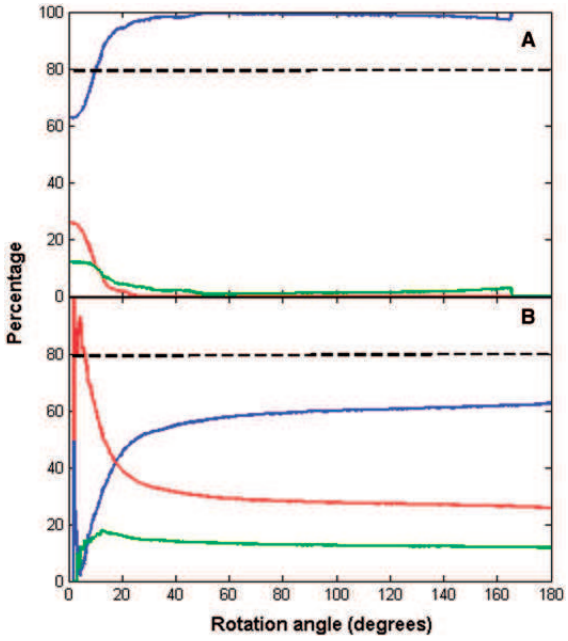

Fig. 6. Predictive value of angle of rotation. Blue lines correspond to 'Hinge', green lines to 'Mixed' and red lines to 'Shear'. (**A**) A point on a line gives the proportion (in percentage) of domain movements (excluding non-contact cases) with rotation angles greater than or equal to that given at the point, that are from the set indicated by the colour of the line. (**B**) A point on a line gives the proportion (in percentage) of domain movements (excluding non-contact cases) with rotation angles less than that given at the point, that are from the set indicated by the colour of the line

domain movements (excluding non-contact cases) with rotation angles of $<6^\circ$, 80% are shear.

3.3 Translation in domain movements

If the shear concept relates to translational movement, then one would expect a large proportion of the shear set to have an interdomain screw axis located outside the body of the protein. However, of the 361 shear examples, only five (1.4%) have an axis outside the body of the protein (using a cut-off distance of 5.5 Å between the axis and any heavy atom of the protein). For the 884 hinge examples, 9 (1.0%) have an axis outside the body of the protein. The rarity of axes located outside the body of the protein indicates that translational movements are rare overall. If there is any truth in the concept of shear indicating a translational movement and hinge indicating a rotational movement, then at least one would expect there to be significantly more cases of remote axes in the shear set than the hinge set. Significance testing on this gave a z-value of 0.56, which gives $p(z \geq 0.56) = 29\%$ for the probability that this difference (1.4% versus 1%) or greater occurs by chance. This result suggests that shear movements are just as likely to have a rotational axis within the body of the protein as hinge movements, implying local control, and that shear movements do not involve the relative translation of one domain relative to the other at least without a rotation occurring about an axis within the body of the protein, i.e. translation is in the axis direction. Considering translation in the axis direction, the mean absolute value for the hinge set is 1.47 Å (SD = 3.1 Å), whereas for the shear set the mean is 0.35 Å (SD = 0.37 Å). Thus, there is significantly more translation along the axis in the hinge set than the shear set, but this is likely to be because of the fact that the rotations are larger among the hinge set. Comparing the pitch would make more sense. The mean absolute value of the pitch for the hinge set is 0.043 Å/degree (SD = 0.095 Å/degree), whereas for the shear set the mean is 0.044 Å/degree (SD = 0.058 Å/degree). Again the difference is not significant ($P = 58\%$).

We also have tested whether the shear set is significantly more likely not to have an effective hinge axis compared with the hinge set. For shear, 61 examples do not have an effective hinge axis (16.8%), whereas the corresponding value for hinge is 117 (13.2%). With a $P = 4.7\%$, this would be significant at the 5% level and suggests that for shear, interactions at the preserved domain interface help control the domain movement, whereas in hinge, it is more likely to be the backbone connections between the domains.

3.4 Twisting movements

The presence of exchanged-partner contact changes is a strong indicator for a shear movement. In our previous work, it was argued that when this type of contact change occurs in isolation, then under certain assumptions concerning the shape of the domains and the location of the hinge axis, this is most likely to occur via a 'sliding twist' movement. A new contact change or an exchanged-pair contact change would most likely occur via either an open-closed or see-saw domain movement. These movements would be closure movements under the same assumptions. This would suggest that twisting movements are more likely to occur in the shear set than the hinge set. For shear, 114 have a predominantly twisting movement (32.0%), whereas the corresponding value for hinge is 192 (21.7%). With a $P = 0.012\%$, this difference is highly significant, showing that twisting movements are more prevalent in the shear set.

3.5 Website

We have produced a website (see http://www.cmp.uea.ac.uk/dyndom/interface) where the domain movements are organized according to whether they are in the no-contact, shear (called 'Interface-preserving movement', see Discussion section), hinge (called 'Interface-creating movement') or mixed set. Each class comprises a list of protein names together with a pair of PDB accession codes and chain identifiers that specify the domain movement. The link provided takes one to a page where the molecular graphics applet, Jmol (http://jmol.sourceforge.net/), is used to display the movement and to indicate the residues that make contact in each conformation. There is also a link to the corresponding DCG classification page and the DynDom page for that domain movement which gives details on the residues comprising the domains, the location of the hinge axis, the hinge-bending residues, the angle of rotation, percentage closure, as well as many other details. A link to the DynDom family page is also provided, which gives a conformational analysis of closely related structures and their domain movements.

4 DISCUSSION

The concept of hinge and shear mechanisms in domain movements was introduced nearly 20 years ago. Assignments of domain movements to these mechanisms were made by an intuitive method that is necessarily subjective. This has limited its application to a small number of domain movements. In the past 20 years, the PDB has grown 30-fold in size and with it the number of implied domain movements. The NRDPDM database contains 2035 unique domain movements, and it would be an onerous task to analyse all of these conformational pairs using molecular graphics software, for the purpose of assigning hinge and shear mechanisms. Therefore, an objective, quantitative method that can be implemented computationally for rapid assignment is needed. The difficulty in achieving this lies in the translation of a subjective method to a quantitative method. There are two pieces of information we can use for this purpose: the description of the subjective method used, and the actual assignments themselves. The description suggested that quantities based on the number of instances in each of the four types of residue contact changes from our previous work (Taylor *et al*., 2013) could be used in distinguishing between preserved interfaces and interface creation. The assignments themselves were used as training data to combine these quantities using logistic regression so as to optimally reproduce the original assignments. The results suggest that we have indeed succeeded in creating a quantitative method for computational assignment of hinge and shear mechanisms. Using this approach, we have managed to classify a much larger set of domain movements into hinge and shear resulting in a 10-fold increase in the number of examples over the number previously available with a high degree of precision.

The term 'shear' and the figures used to illustrate the shear mechanism have led many to interpret a domain closure to occur via a relative translation of one domain relative to the other. Although this is possible, our results have shown that this is rare overall, and no more likely to occur among the shear set than the hinge set. We suggest that the term 'shear movement' is better referred to as 'interface-preserving movement' and 'hinge' as 'interface-creating movement'. These more prosaic terms are still broadly consistent with the original concept but should not lead to misinterpretation.

Our analysis has shown that for proteins with domain movements classified as shear, the movement does not involve a significant translation of the two domains but a rotation about an axis within the body of the protein just as for a protein undergoing a domain movement via the hinge mechanism. We have shown that maintained and exchanged-partner contact changes are strong indicators for shear, whereas exchanged-pair and new contact changes are strong indicators for hinge. The finding that there are significantly more twisting movements in the shear set than in the hinge set is consistent with the notion that a twisting movement can preserve the domain interface. This offers one explanation of how a rotational movement can preserve an interface without relative translation. However, not all predominantly interface-preserving movements occur via a twisting motion; many can still occur via a closure motion by rotation about well-defined hinges.

The case of citrate synthase illustrates how a 'predominantly shear' movement as designated by DBMM would still be appropriately described as hinge-bending even though it is in our mixed class (prediction value of 0.55) with slightly more interface-preserving features than interface creating. Figure 2A shows the DCG for citrate synthase. There are 10 maintained contact changes, 2 exchanged-partner contact changes, 2 exchanged-partner contact changes and 6 new contact changes. It has a well-defined hinge axis created by mechanical hinges, one of which is a 'hinged-loop' (Hayward, 1999), a loop flanked by two bending regions through which the hinge axis passes. This hinged-loop clearly helps control the domain movement just as a hinge would in a protein conventionally regarded as undergoing closure via hinge bending, e.g. lactoferrin. The domain movement in citrate synthase is also an example of a protein that undergoes closure (84%) via hinge bending, but one that preserves some part of the domain interface.

ACKNOWLEDGEMENTS

The authors thank Russell Smith for help with the construction of the website.

Conflict of Interest: none declared.

REFERENCES

Abrahams,J.P. *et al*. (1994) Structure at 2.8 Å resolution of F1-ATPase from bovine heart-mitochondria. *Nature*, **370**, 621–628.

Amemiya,T. *et al*. (2011) Classification and annotation of the relationship between protein structural change and ligand binding. *J. Mol. Biol*., **408**, 568–584.

Bennet,W.S. and Huber,R. (1984) Structural and functional aspects of domain motions in proteins. *Crit. Rev. Biochem*., **15**, 291–384.

Bennett,M.J. *et al*. (1994) Domain swapping: entangling alliances between proteins. *Proc. Natl Acad. Sci. USA*, **91**, 3127–3131.

Berman,H.M. *et al*. (2000) The protein data bank. *Nucleic Acids Res*., **28**, 235–242.

Brylinski,M. and Skolnick,J. (2008) What is the relationship between the global structures of apo and holo proteins? *Proteins*, **70**, 363–377.

Cawley,G.C. *et al*. (2007) Generalised kernel machines. In: *2007 IEEE International Joint Conference on Neural Networks*. Vol. 1–6, Orlando, Florida, USA, pp. 1720–1725.

Cawley,G.C. and Talbot,N.L.C. (2008) Efficient approximate leave-one-out cross-validation for kernel logistic regression. *Mach. Learn*., **71**, 243–264.

Cessie,S. and Houwelingen,J. (1992) Ridge estimators in logistic regression. *J. R. Stat. Soc. Ser. C Appl. Stat*., **41**, 191–201.

Chasles,M. (1830) Note sur les propriétés générales du système de deux corps semblables entr'eux et placés d'une manière quelconque dans l'espace; et sur le déplacement fini ou infiniment petit d'un corps solide libre. *Bull. Sci. Math*., **14**, 321–326.

Eklund,H. *et al*. (1981) Structure of a triclinic ternary complex of horse liver alcohol-dehydrogenase at 2.9 a resolution. *J. Mol. Biol*., **146**, 561–587.

Gerstein,M. and Krebs,W. (1998) A database of macromolecular motions. *Nucleic Acids Res*., **26**, 4280–4290.

Gerstein,M. *et al*. (1994) Structural mechanisms for domain movements in proteins. *Biochemistry*, **33**, 6739–6749.

Hayward,S. (1999) Structural principles governing domain motions in proteins. *Proteins*, **36**, 425–435.

Hayward,S. and Berendsen,H.J.C. (1998) Systematic analysis of domain motions in proteins from conformational change: new results on citrate synthase and T4 lysozyme. *Proteins*, **30**, 144–154.

Hayward,S. and Lee,R.A. (2002) Improvements in the analysis of domain motions in proteins from conformational change: DynDom version 1.50. *J. Mol. Graph. Model*., **21**, 181–183.

Qi,G. *et al*. (2005) A comprehensive and non-redundant database of protein domain movements. *Bioinformatics*, **21**, 2832–2838.