J Comput Aided Mol Des (2009) 23:773–784
DOI 10.1007/s10822-009-9273-4

Identification of family-specific residue packing motifs and their use for structure-based protein function prediction: I. Method development

Deepak Bandyopadhyay · Jun Huan · Jan Prins · Jack Snoeyink · Wei Wang · Alexander Tropsha

Received: 21 July 2008 / Accepted: 15 April 2009 / Published online: 20 June 2009

Abstract Protein function prediction is one of the central problems in computational biology. We present a novel automated protein structure-based function prediction method using libraries of local residue packing patterns that are common to most proteins in a known functional family. Critical to this approach is the representation of a protein structure as a graph where residue vertices (residue name used as a vertex label) are connected by geometrical proximity edges. The approach employs two steps. First, it uses a fast subgraph mining algorithm to find all occurrences of family-specific labeled subgraphs for all well characterized protein structural and functional families. Second, it queries a new structure for occurrences of a set of motifs characteristic of a known family, using a graph index to speed up Ullman's subgraph isomorphism algorithm. The confidence of function inference from structure depends on the number of family-specific motifs found in the query structure compared with their distribution in a large non-redundant database of proteins. This method can assign a new structure to a specific functional family in cases where sequence alignments, sequence patterns, structural superposition and active site templates fail to provide accurate annotation.

Keywords Structural genomics · Protein graphs · Protein function prediction · Family-specific motifs · Frequent subgraph mining · FFSM

Electronic supplementary material The online version of this article (doi:10.1007/s10822-009-9273-4) contains supplementary material, which is available to authorized users.

D. Bandyopadhyay (✉)
GlaxoSmithKline, 1250 S. Collegeville Rd, Mail Stop UP12-210, Collegeville, PA, USA
e-mail: Deepak.2.Bandyopadhyay@gsk.com

J. Huan
Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA
e-mail: jhuan@eecs.ku.edu

J. Prins · J. Snoeyink · W. Wang
Department of Computer Science, University of North Carolina, CB#3175 Sitterson Hall, Chapel Hill, NC, USA

J. Prins
e-mail: prins@cs.unc.edu

J. Snoeyink
e-mail: snoeyink@cs.unc.edu

W. Wang
e-mail: weiwang@cs.unc.edu

A. Tropsha (✉)
School of Pharmacy, University of North Carolina, CB#7360 Beard Hall, Chapel Hill, NC, USA
e-mail: alex_tropsha@unc.edu

Introduction

Protein functional annotation is an important focus of molecular biology that has critical implications for drug discovery. Protein targets of known drugs come from just over 120 different families [1], from among about 1000 unique protein folds [2], and thousands of unique protein functions. Finding and characterizing new targets can greatly expand our ability to identify novel drugs.

One rich source for new targets are the genome sequencing [3] and structural genomics [4] projects, which have produced a plethora of new protein sequences and structures, respectively. A significant fraction of the protein-coding sequences from the genome projects corresponds to proteins that have not been characterized experimentally, called hypothetical proteins [5]. Likewise,

a large percentage of structural genomics targets deposited in the Protein Databank (PDB) lack experimental functional annotation. Often, inferring the function of a new protein as similar to that of a known protein with similar sequence or fold (global structure) is problematic or misleading [6].

Structure determination by structural genomics outpaces the rate of experimental function characterization. There are growing numbers of orphan structures, i.e. proteins with unknown function and no apparent homology with known functionally characterized proteins. An examination of 1600 proteins from structural genomics projects that were deposited between January 1999 and April 2005 (Fig. 1) indicates that only about 50% of them were assigned a functional annotation, and another 25% could be assigned a function with high confidence using global structural similarity [7]. That leaves 25% (382 out of the 1600) orphan structures, whose PDB IDs are listed in the Supplementary Material. Innovative, non-conventional computational approaches are needed to infer the function of such orphan protein structures.

Recently, we have begun to address this problem using graph representations of protein structure [7, 8] and frequent subgraph mining algorithms [9]. In this paper, we present novel methodological developments that enable the rigorous identification of protein family-specific residue motifs. In addition, the accompanying paper [10] discusses several examples of method application. Before covering our method in detail, we shall discuss briefly previous efforts in the field, covered in more detail in recent reviews [11–13].

Related work

Successful computational methods for predicting protein function tend to be knowledge based; i.e., they use information derived from proteins with known function to annotate similar or related proteins. These methods fall into three broad categories based on the type of similarity they exploit: sequence similarity, overall structural similarity, or local structural similarity. In addition there are integrative methods that assign function by combining functional information from different sources. Here we focus on functional annotation using local structural data, and briefly mention other methods below.

Annotation methods based on sequence similarity

Functional annotation based on sequence alignment is possible when one can identify another protein or domain of known function with at least 40% sequence identity to a query protein. 99% of the protein pairs with sequence identity above 40% have similar structure, and more than 90% of protein pairs with more than 70% sequence identity have the same function [14, 15].

Sequence-based functional annotation is challenging in the absence of reliable sequence similarity. Some effective methods include *sequence patterns*, regular expressions derived from sequence alignments [16]; and *sequence profiles*, probabilistic regular expressions containing frequencies of amino acid occurrence at each sequence alignment position (PSSM [17], PSI-BLAST [18]). *Hidden Markov Model* (HMM [19]) profiles rigorously convert a

(a)

Year	#SG dep	unkn fn	DALI z<12
99	11	5	3
00	44	7	5
01	88	33	20
02	146	59	37
03	436	189	101
04	723	320	186
05*	157	56	30
total	1605	669	382

(b)

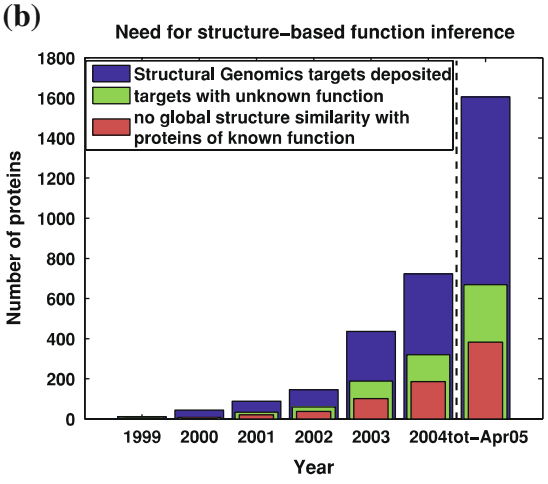

Fig. 1 Number and functional classification of structural genomics targets, 1999–2005. Number of structural genomics targets released each year from 1999–2005 (*April), split into those with known and unknown functions, shown in (a) table and (b) graph form. The proteins with unknown function (∼50%) are further split into those whose function may be inferred on the basis of strong global structural similarity to proteins of known function (DALI z-score > 12), and those with no strong global structural similarity. The last column/row "total" is cumulative

multiple sequence alignment into a profile by determining transition probabilities between residue match, insert and delete states at each position in the alignment. HMM profiles can find remotely related sequences [20].

Phylogenetic and evolutionary methods for detecting sequence similarity model the changes in protein sequences assuming their divergence from each other during evolution. Some methods include evolutionary traces [21–23] and comparative genomics [24] methods such as Clusters of Orthologous Groups [25], gene co-evolution [26], and cross-species co-occurrence [27].

Recent advances in sequence similarity searching include graph models of residue coupling, i.e. residue pairs that change together in an alignment [28]; neighborhood correlation in the sequence similarity network, which accommodates multidomain proteins and domain swapping [29]; and Protein Function Templates (PFT/LIMACS, [30]) which include quantitative information about functional sites in their multiple alignment and profile/PSSM to yield better annotation. ModFun [31] adds similarity of protein interaction partners to improve the specificity of sequence annotation with PSI-BLAST.

Annotation methods based on global structural similarity

It is generally agreed that protein function is defined by its structure, which is better conserved during evolution than sequence [32]. Overall structural similarity to proteins of known function may thus offer clues about the function of an orphan protein, especially for remote sequence homologs. Proteins with overall structural similarity at different levels are grouped into hierarchical classifications such as SCOP [33] and CATH [34]. Fast algorithms for pairwise structure comparison have been developed [2, 32, 35, 36]. For instance, DALI [2] assigns a z-score to each structural match based on the probability of two random structures having a match with the same RMSD and length of alignment; 90% of protein pairs with DALI z-score above 12 were shown to have the same function [37].

Annotation methods based on local structural similarity

In the absence of sequence and global structural similarity, local structural patterns, often called *residue packing patterns* or *structural motifs*, often give important insights into function. The hypothesis that protein function is determined not by overall fold but by a few functionally important residues is supported by convergent evolution of function, loss of function upon mutation of key residues, and the diversity of folds for some protein functions [38].

Below we review methods employing local structure comparison, as opposed to ones that map sequence motifs onto structure [6, 39, 40]:

- *Depth-first search* starts from simple geometric patterns such as triangles, and progressively finds larger patterns. This method was first used to find local side-chain packing patterns by Russell [41]; this group subsequently developed a method to find binding site patterns in non-homologous structures (PINTS, [42]) and applied it to structural genomics proteins [43]. TRILOGY [44] looks for patterns among conserved residues within a family, combining separate sequence and structure matches, and building longer matches from smaller ones. Similarly, Med-SuMO compares functional sites based on patterns from triplets of chemical groups surrounding ligand binding sites [45].
- *Geometric hashing*, which compares objects through hashed coordinates, has been used to compare two protein structures [46], compare a structure to a database [47], and find functional sites in structural genomics proteins [48].
- *Functional site template* methods represent known functional sites as pockets [49], clefts [50], or patches [51], and match new protein structures using geometry, conserved residues and electrostatic/chemical properties.
- *String pattern matching* uses string search algorithms on encoded local structure and sequence [41, 52].
- *Graph matching* methods have been developed to compare protein structures modeled as graphs, usually with clique detection techniques. Most of the techniques [36, 41, 53–57] search using graph representations of existing functional sites, while a few [58, 59] mine these from protein families.
- Other methods for inferring motifs from protein 3D structure include inductive programming language [60], fuzzy functional forms [61], computed protonation properties [62], and geometric depth potentials [63].
- *Hybrid methods* Some methods combine the benefits of different approaches, such as geometric hashing to speed up clique detection [64].

Consensus methods

Often, different function prediction methods give conflicting clues, and one would prefer to arrive at a consensus based on the relative confidence of each prediction, or just provide a few alternative functional assignments.

ProFunc [65] employs a consensus of different sequence, structure and functional site methods to infer protein function. Sequence methods employed include BLAST [18], InterPro [66] and Superfamily [67]. SSM [36] is used for fold match detection, and functional site methods include Relibase ligand templates [68], Catalytic Site Atlas [69],

DNA-binding motifs [70], nests [71], and reverse templates [72]. Results from different analyses are either presented separately, or combined into a consensus prediction [73].

Materials and methods

Our algorithm includes five steps split between two major components: family motif identification (steps 1–3) and function prediction for a query structure (steps 4–5). The first three preprocessing steps are run once for each selected family, producing a motif library against which new structures can be scanned.

Steps 4–5 are run for each query structure, and each prediction is characterized with a confidence value. These five steps are as follows:

1. *Select families* of non-redundant proteins from any classification scheme such as SCOP or EC, or as defined by the user. Also, define the *background* dataset that will represent all remaining protein structures.
2. *Represent protein structures as graphs*, with nodes at the C_α atom of each residue (residue name is used to label a node), and contact between residues defined using the *almost-Delaunay* [74] edges. This set expands the set of geometric nearest neighbors to include pairs of points that could be nearest neighbors if points were allowed to move from their defined coordinates by up to ε, thus accounting for imprecision in atomic coordinates. Our recent work [8] showed that almost-Delaunay edge graphs are sparse and robust enough to find complex patterns from protein families quickly in the presence of coordinate perturbations.
3. *Mine family-specific motifs* using the Fast Frequent Subgraph Mining method [8]. Motifs are defined as family-specific if they occur in at least 80% of the family (*support*), and at most 5% of the background (*background occurrence*).
4. *Search for motifs* in a structure to be annotated, using an index of graph similarity to speed up Ullman's subgraph isomorphism [75].
5. *Assign a significance* to the function inference from the number of motifs found and its distribution in background proteins.

Availability Steps 1–3 constitute the FFSM software described previously [8], which is implemented in C++ and Perl, and is available from http://www.cs.unc.edu/huan/FFSM.shtml. Steps 4–5 for function inference and characterization are implemented in MATLAB and available in the ADMatlab bundle released by the first author at http://www.cs.unc.edu/debug/software.

Family and background selection

We selected families from the SCOP structural classification and the EC functional classification (Enzyme Commission [76]). One could also use other classifications such as Gene Ontology [77] or COGs [25], or manually selected groups of proteins.

The background dataset is a non-redundant subset of the PDB used to check the specificity of frequent patterns mined from families. It was selected by downloading from PISCES [78] the precomputed CulledPDB dataset with maximum 90% sequence identity, better than 3 Å resolution, and R-factor at most 1.0. This led to a set of 6749 protein chains when this analysis was first done (on May 29, 2004); at the time of writing the same parameters produced a dataset of over 13,000 chains.

SCOP families were downloaded from version 1.65 of the database, which was current when we initiated these studies; the later version 1.67 was used to validate the method as described in the companion paper [10]. EC families were obtained using the Thornton group's PDB to EC mapping.[1] Families in EC were removed if a SCOP family was found to have exactly the same set of non-redundant members. Thus, EC families are retained only when they represent functionally related proteins that are scattered over different SCOP families (e.g. halocompound dehalogenases, muconate lactonizing enzymes, amino acid racemases), or when they add many new members to a family that is poorly represented in SCOP 1.65 (e.g. shikimate dehydrogenase, dehydroquinate dehydratase).

Non-redundant lists of family members were created by intersecting protein chains classified under a SCOP node or an EC number with the background dataset. This avoided inclusion of nearly identical structures that may bias the family composition and also invoke the worst case exponential behavior of subgraph mining.

Modeling protein structures by graphs

We represent protein structures using graphs, with nodes at each residue labeled with the amino acid type, with V,A,I,L merged into a single type since they often substitute for one another. The other hydrophobic residues and other frequently substituted pairs such as D and E are kept distinct to detect patterns of their conservation. The *almost-Delaunay* edges [74] define contacts between residues in the graph. We augment each contact by the Euclidean distance between the two residues the edge connects [79]. Graphs built using almost-Delaunay edges were previously shown

[1] Enzymes database http://www.ebi.ac.uk/thornton-srv/databases/enzymes, and flat file downloaded from http://www.ebi.ac.uk/thornton-srv/databases/pdbsum/data/seqdata.dat.

to afford faster and more accurate mining of frequent patterns than graphs with residue contacts defined using either Delaunay edges or a distance threshold [8]. Edges are labeled with length ranges (0–4, 4–6, 6–8.5, 8.5–10.5, 10.5–12.5 and 12.5–15 Å). Motifs mined using this graph representation are called *weighted edge motifs*, as opposed to unweighted edge motifs described in our previous study [80] where we used only two edge labels (sequence-adjacent or spatially-proximal) vs. edge length labels in the present study.

All the proteins in the selected families were converted into their graph representations, as were all proteins in the background. There are 6515 protein chains in the background dataset for the unweighted edge representation, but 6625 for the weighted edge representation, because an improved PDB parser allowed us to process more proteins.

Mining family-specific motifs

We mine frequent subgraphs from the graph representations of multiple proteins using the FFSM algorithm [9]. Some parameters are set for the mining step, based on the characteristics of the families being mined. These include the *minimum support* (f, default 80%) and *maximum background occurrence* (b, default 5%) that were described previously [80]. We introduce the following new parameters:

- *Maximum size* (s) of a frequent subgraph to report is set by default to eight residues. While larger subgraphs are useful while studying the biological relevance of the motifs [8], smaller subgraphs are desirable for the purpose of function inference, since they can accommodate more variation across the family and find members with slight variations in the geometric arrangement of the residues comprising the motif. To allow such variations, it is also necessary to report all frequent subgraphs rather than only maximal frequent subgraphs [81], to ensure that smaller subsets of family-specific motifs that are themselves family-specific are available as features for function inference.
- *Minimum subgraph density* (d) is expressed as the maximum number of edges that one may add to the reported subgraphs to make them cliques. This parameter serves to control the rigidity and interconnectedness of frequent subgraphs and motifs. By default it is set to three edges missing from a clique for unweighted edge motifs, and one edge missing for weighted edge motifs.

The default settings of the parameters specified above generate reasonably dense and biologically relevant subgraphs for most families in less than 2 h on a 2.8 GHz Linux machine with 1 GB of memory. Any frequent subgraphs that occur in more than a fraction b of the background are removed from consideration, and the remaining subgraphs are stored as the family motifs. The default values of mining parameters produce a reasonable number of motifs (between 10 and 1000) for many families; for very small or heterogeneous families, the parameters must be varied to attain this target number, as described later in the Results on Family classification.

Querying a new protein using a graph index

The problem of annotating a query protein with a set of motifs from a candidate family can be posed as searching the graph of the query protein for occurrences of family-specific subgraphs. This subgraph isomorphism search problem is known to be NP-complete [75]. To make it tractable, we build indices for each motif and for the query structure containing some precalculated information about the graphs. If by comparing indices one finds that a query structure cannot contain a particular motif, one can stop the search without checking for subgraph isomorphism.

We have devised a graph index called a *local neighborhood* index, where for each node in a motif we count the number of occurrences of nodes with different labels reachable on paths with increasing lengths starting from that node, as shown in Fig. 2. An index match occurs when these counts for each node in a motif are equal to or exceeded by a node with the same label in a query protein, followed by a graph isomorphism search to confirm the occurrence. Further technical details on calculating graph indices can be found in the Supplementary Material.

Assigning significance to a function inference

Suppose there are C_F motifs for family F, $X_1 \ldots X_{C_F}$ with family support f (say 0.8) and background occurrence b (say 0.05). Suppose a set of these motifs $X_{q1} \ldots X_{qn}$ are found in a query protein q by the subgraph isomorphism algorithm above. One may then simply count the motifs found to get a good first estimate of the confidence of q belonging to F. Each motif occurs in a fraction of at least f family members, and of at most b background proteins. Then, were the occurrences of these motifs to be independent and normally distributed, one would expect to see on average $f\ C_F$ motifs in a family protein and $b\ C_F$ in a background protein, normally distributed about these means; the number found in a query protein could be looked up in these distributions to determine $P-value$ of family or background membership.

However, the number of motifs in a background protein is not normally distributed but looks like a Poisson distribution, as shown in Fig. 3. Individual motifs are also not independent, since some of them share residues, overlap,

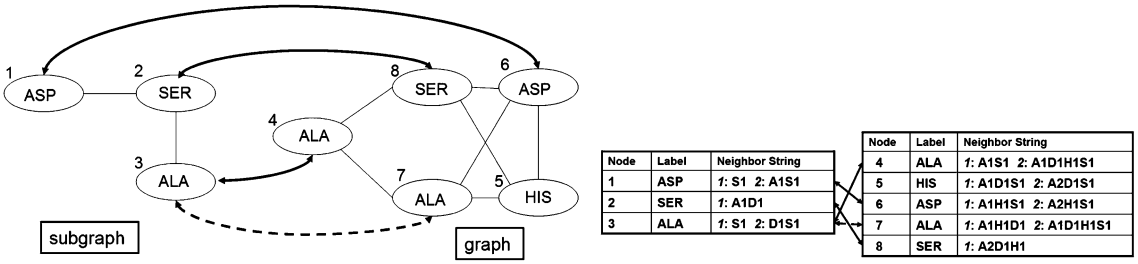

Fig. 2 An example of the local neighborhood graph index. An example of the local neighborhood graph index (shown for unweighted edge graphs; ALA could be V/A/I/L). *Left*: Subgraph (motif) matched to graph (query protein). Solid arrows connect two nodes with matching indices in motif and query; a dotted arrow connects two nodes with the same label but with incompatible graph index vectors, that cannot be part of a successful match. *Right*: node ID, node type, and index (neighbor string) for each node in the two graphs, with the same nodes connected by solid and dotted arrows

and cover the same set of residues in different ways. Thus, we do not use the $P-value$ for family membership from a Poisson distribution, but instead compute it empirically as discussed below.

Sensitivity and specificity based on number of motifs

Whereas family specific motifs are defined based on their general prevalence within family members as compared to the background set, the issue of inferring function for a single protein is based on the number of family motifs present within this protein. Thus, we shall seek to define a cutoff value for the number of a family specific motifs that a protein is required to have in order to infer its family membership.

For any value picked as cutoff, the *sensitivity* is defined as the fraction of known family members with at least that many motifs. The *specificity* is defined as the fraction of all proteins in the background set (i.e. not in the family) that have fewer motifs than the cutoff, and hence are correctly inferred as not being family members. Specificity can also be defined as 1 minus the false positive rate. Too low a cutoff leads to false positive hits, while too high a cutoff misses family members that do not contain all or most of the motifs. The optimal cutoff depends on motif mining parameters (support and background frequency) and characteristics of the family itself, such as its size and homogeneity, that affect the distribution of motifs in the family and background. We define ROC curves for family membership prediction, plotting specificity and sensitivity of inferring function by selecting proteins from the background having different numbers of motifs (from none to all). As an example, Fig. 3 shows the number of unweighted and weighted edge motifs within the Immunoglobulin light (V) chain family (370 non-redundant proteins) and within the background (6255 non-redundant proteins) in histograms, with ROC curves superimposed.

It is desirable to choose a cutoff with some minimum specificity (e.g. 95, 99%), while allowing a limited loss in sensitivity to accommodate outliers in SCOP (super-)families. Motifs that are good predictors would reach close to 100% specificity and sensitivity at some value of the

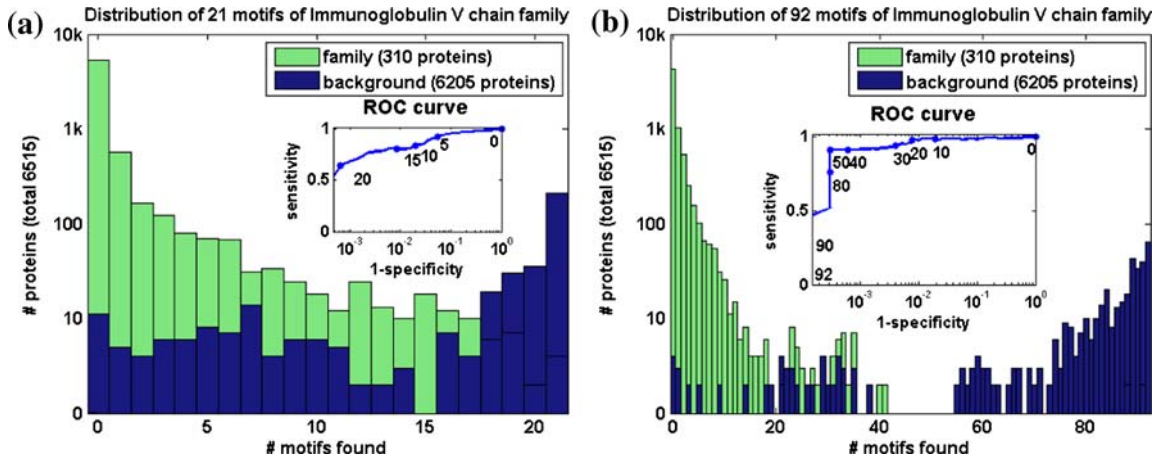

Fig. 3 Background distribution and ROC curve of Immunoglobulin V-chain motifs. Distribution of Immunoglobulin V-chain motifs in the background (light green), and within the family (dark blue, superimposed in front), shown for (**a**) unweighted and (**b**) weighted edge motifs. *Inset*: ROC curves showing specificity vs. sensitivity of function inference at different cutoff points

cutoff, and such a *knee point* can be chosen as a cutoff. However, often there are multiple knee points and it is desirable to have an automated procedure to pick the best cutoff for a range of families with differently shaped ROC curves. Also, it is sometimes necessary to conduct two kinds of searches using motifs, a *sensitivity-biased* search that ensures one does not miss family members, and a *specificity-biased* search when scanning a huge database of targets with the objective of turning up the most promising function inferences while avoiding spurious hits. It makes sense to have two different cutoff values for these two searches.

The strict 99%-specificity cutoff point is set to the number of the family motifs that occur in no more than one percent (ca. 60 proteins) in the background set. This cutoff point ensures 99% specificity regardless of sensitivity; thus for some families less than half the known members can be inferred with 99% specificity. For protein families where even having all the motifs does not ensure 99% specificity, the 99%-specificity cutoff point is set at the total number of motifs, and family-specific motif-based function inference is expected to be unreliable.

The *sensitivity cutoff point* aims to approximate a knee point of the ROC curve, while ensuring at least 95% and no more than 99% specificity. The heuristic for sensitivity cutoff starts by finding the first two points of loss in sensitivity, i.e. the proteins with the two lowest motif counts within the family. Then we select the lower one of these points that exceeds 98% specificity; if none of them do, we select the lower one that exceeds 95% specificity. If both points of loss in sensitivity have below 95% specificity, we discard them and choose the 95%-specificity point as the sensitivity cutoff point. Similarly, if both points of loss in sensitivity have greater than 99% specificity, as happens in families where all members contain all the motifs, we pick the sensitivity cutoff point the same as the 99%-specificity cutoff point.

Several proteins in large families with 10–40 non-redundant members have too few motifs to be reliably inferred even with 95% specificity. Often this is because of functional divergence within a SCOP family or superfamily, where the motifs capture the predominant function, and members with diverged functions are detected as having fewer motifs. However, sometimes this can indicate errors in the PDB files, which can be resolved by cleaning the motifs as described in the Supplementary Material. Conversely, families with < 4 members, < 15 motifs, family support (f) $< 70\%$ or background occurrence (b) $\geq 15\%$ usually have a broad histogram of motif occurrence in the background, which forces a higher cutoff point. This is usually accompanied by loss in sensitivity; in other words, few family members have enough motifs to reliably infer their function.

Results

We shall list the SCOP and EC families for which we have derived motifs with our method, and categorize them based on their size and the number and nature of their motifs.

How many families yield motifs?

In all, among 182 families from SCOP and 46 from EC mined for motifs, 120 SCOP and 27 EC families yielded a sufficient number of unweighted edge motifs, and 125 SCOP and 26 EC families produced weighted edge motifs. The fraction of SCOP families that have a sufficient number of motifs is 66%. Since the family and superfamily level nodes we selected are nearly randomly distributed within SCOP, this fraction gives an estimate of the percentage of SCOP (super-)families whose members share a single function. The fraction is lowered by the large number of heterogeneous superfamilies that end up not having enough motifs and are omitted.

The fraction of EC families with sufficient number of motifs is even lower at 61%, though EC is a functional classification. We propose several reasons—broad families at top levels of the EC hierarchy whose members share only broad characteristics of a function, and not its mechanism; functional sites accommodating substitution of amino acids that we currently treat as distinct (e.g. D and E, or F and W); functions spanning several folds and with large differences in their active site geometry [38]; and mechanistically diverse families (e.g. the enolase superfamily) known to pose a challenge for function inference [82]. The fraction of EC families that yield motifs (61%) is a rough estimate of the percentage of enzyme functions in EC with the same local structure and mechanism. The estimate is rough since we did not work with a representative sample of EC; also, the EC system's historical inconsistencies and other shortcomings make it not well suited for function inference [83].

Family classification based on motif mining parameters

We may get different numbers of motifs for the same family by adjusting the values of three parameters: family support (f), background occurrence (b) and graph density (d). Thus we require the number of motifs for each family to be in a target range of 11–999 (20–200 preferred). These limits are based on empirical observations: using 10 or fewer motifs one cannot distinguish true and false positives with confidence. On the other hand, 1000 or more motifs are redundant for typical protein families; they cover the same 100–500 residues in different ways, and overfit the sample of the protein family selected for motif mining, precluding identification of other proteins having similar

functions. Hence, we tune the values of parameters *f*, *b* and *d* to achieve this target number of motifs for different families. It is useful to classify families based on the mining parameters used to obtain the target number of motifs:

- *Ideal families* have at least 5 members, most of which share the same function, but their sequences and structures differ enough that only a few motifs may be found and they correspond to functionally important residues. This is the family type that our subgraph mining algorithm with default parameters of *f* 0.8, *b* 0.05 and *d* 1 (unweighted edges) or 3 (weighted edges) works best on. In fact, changing the parameters does not affect the motifs from many of these families since there is a definite set of motifs, all highly specific to the family. For example, eukaryotic serine protease is an ideal family.
- *Rich families* have at least 5 members that are very similar in sequence and structure as well as function, and thus have too many motifs. Parameter values for rich families should be restrictive to bring the number of motifs below 1000—typically *f* is 0.9 or 1.0, *b* is 0.05, 0.02 or 0.01, and *d* is 1 or 0. For example, TIM (triosephosphate isomerase) is a rich family.
- *Poor families* have at least 5 members that share the same function, but differ enough in structure and/or sequence that they share no (or very few) motifs when mined with the default parameters. Parameter values for poor families should be permissive—typically *f* is chosen as 0.7 or 0.8, *b* is 0.1 or 0.15, and *d* is 2 or 3. For example, α/β-knot methyltransferase is a poor family.
- *Functionally heterogeneous families* have at least 5 members, and at least two different functions are well represented in their members. Heterogeneous families again need to be mined with permissive values of parameters, e.g. setting *f* 0.7. For example, some superfamilies of TIM barrel fold (Ribulose phosphate binding barrel, metallo-dependent hydrolase) are heterogeneous families.
- *Tiny families* have 3–4 members. Wangikar et al. [58] had observed that families with less than 5 proteins may have few frequent patterns that are significant, i.e. have discriminating power. Patterns that are frequent in two out of three or three out of four proteins usually run into the hundreds of millions, and are mostly spurious and not specific to the family. Thus, we use extremely restrictive values of the mining parameters for tiny families: *f* is 1.0 (patterns must occur in all family members), *b* is 0.05, 0.02 or 0.01 (highly specific), and *d* is 0 or 1 (highly connected/dense). With these choices we are able to mine most tiny families for motifs, whose discriminating power we confirm in the companion paper [10]. For example, the Sec7 domain is a tiny family with 3 members.
- *Omitted families* are either heterogeneous, poor or tiny families that did not yield at least 10 motifs using permissive mining parameters (e.g. S-adenosine-L-methionine (AdoMet) dependent methyltransferase, SCOP: 53335), or rich families that had over 1000 motifs even with restrictive parameters (e.g. isopropylmalate dehydrogenase, EC 1.1.1.85, had 3332 weighted edge motifs even with *f* 1.0, *b* 0.01 and *d* 0). Both these were omitted from further analysis, as were families with less than three non-redundant members.
- *Augmented families* are those that were omitted, tiny or poor families using the non-redundant members from SCOP 1.65, but can be converted into ideal families by adding new members from SCOP 1.67 or other sources. We could mine motifs only from the SCOP 1.67 version for the CheY-like superfamily (SCOP ID: 52172) and family (52173). Also, we mined motifs from both SCOP 1.65 and 1.67 for haloacid dehalogenases (56784) and antibiotic resistance proteins (54598) that were poorly represented in protein structure space in SCOP 1.65.

All our chosen SCOP and EC family datasets that yielded sufficient unweighted or weighted edge motifs are listed in Tables II–V in the Supplementary Material.

Performance of the graph index

Performance of the graph index is measured by the speed of retrieving results as well as the *efficiency* (fraction of real embeddings per index hit for a single subgraph) and *hit rate* (fraction of real embeddings per index hit in a single query). In the Supplementary Material we prove that our graph index improves the speed, efficiency and hit rate for the subgraph isomorphism search, making it feasible to search for multiple motifs within large families in a reasonable time.

Discussion

Comparative contribution of motif based function inference

We shall compare the relative merits of function inference using structure-based family-specific motifs versus previous efforts, to establish its scope and applicability. Sequence-based functional annotation has been more popular than other approaches since there are an order of magnitude more sequences than structures, and the gap is growing since

genomes are sequenced faster than structures are solved. However, the success of annotation based on sequence alignment depends on a high degree of sequence similarity—30% pairwise sequence identity is considered the lowest threshold for homology modeling, and more than 40% for reliable transfer of function annotation from a known to an unknown protein [84]. Structural genomics targets usually share less than 30% sequence identity with proteins of known function, since they are deliberately selected in this way to maximize coverage of fold space. Thus, they cannot be reliably annotated by sequence alignment.

Sequence patterns/motifs derived from an alignment are usually constrained to occur in the same order within the sequences across all members of a family. Thus, sequence methods may not find patterns of residues that do not follow the conventional sequential order in some members of a family (but are similar in three dimensions). They also fail to classify proteins in which a pattern occurs out of sequential order as belonging to the family. Modifications such as circular permutations [85] lead to homologous proteins not detectable by sequence comparison, while domain insertion/ deletion, strand invasion, and internal swapping of β-hairpins [86] lead to different structures and functions within a family of evolutionarily-related proteins.

Though the majority of structural genomics proteins have low sequence similarity to known proteins, up to 75% of them have enough overall structural similarity to reliably infer function. Frequently, the annotation suggested by global structural similarity reveals incorrect annotation previously assigned by sequence similarity [87]. Several reviews discuss the comparative merits of sequence and structure-based methods for functional annotation and the accuracy of annotation transfer [84, 88–90].

Inference of function from overall structure similarity may be problematic, because similar folds do not necessarily imply a similar function. For example, the TIM barrels are a large group of proteins having a similar fold but very different functions [91]. On the other hand, similar function does not require similar fold. For example, the most versatile enzymes, hydro-lyases and O-glycosyl glucosidases, are associated with 7 folds each [38]. Function assignment based on identification of functional sites was shown to be more accurate than that based on the overall most similar sequence or structure [6], and it may be the only feasible computational means to suggest functions for structural orphans, proteins whose sequence and structure have no similarity to others of known function.

Most methods for function assignment can detect only active site templates from known protein families that have already been characterized in the literature. This precludes the identification of similarity to a hitherto unknown functional site, or cases where some active site residues have mutated or the active site geometry is distorted. Modeling of functional sites by only functionally important charged and polar residues [53, 58] precludes the identification of functions such as membrane proteins or lipid binding, where hydrophobic residues are functionally important. Also, identification of a functional site does not always lead to identification of function, since many families share functional sites (such as ATP or NAD binding sites), and many active sites could carry out more than one function [92]. It is also known that combinations of protein domains often have novel functions, different from the characteristic of the same domain in single domain proteins [90].

Some local structure search methods such as geometric hashing [93] and clique detection [58, 59] have been used to infer recurring spatial motifs from groups of structures. Both methods have exponential running time as the number of structures increases. Our subgraph mining algorithm [9] builds frequent subgraphs of arbitrary topology directly using a tree representation, typically taking less time for larger families than smaller ones. Thus it is faster and applicable to larger structures and databases than exhaustive subgraph enumeration by depth-first search [36].

The calculation of motifs by the method is fully automated and does not assume any prior knowledge of functional sites, though such knowledge can be incorporated to guide or restrict the mining process. In contrast to functional sites, family-specific motifs are highly specific to their families, and thus the discovery of a family motif in a protein of unknown function is more significant than the discovery of local structural similarity to one or a few unrelated proteins that might occur by chance. The use of a robust graph representation and multiple motifs required for a match increase the confidence of function assignment, compared to the identification of a single functional site.

There have been recent efforts towards the annotation of protein structures (and homology models built from sequences) using functional signatures derived from structural alignments [94], overlapping sphere representations of functional sites [95, 96], and clusters of functionally important residues determined by predicted protonation properties [62] or a geometric depth potential [63, 97], to name just a few. Our method, unlike the first [94], does not depend on a sequence or structure alignment, and can find motifs not conserved in the sequence. It differs from the second [96] in that the functionally important residues used in graph patterns are inferred from protein families rather than chosen manually from the literature or bound ligand positions. It distinguishes itself from the other methods mentioned [62, 63] by insisting that motifs found and used for annotation be unique to each family.

The PHUNCTIONER method [98] groups proteins based on GO terms, and creates associated libraries of 3D

profiles storing fragments of aligned structure and their associated function-specific sequence alignments and position-specific scoring matrix (PSSM). PHUNCTIONER chooses conserved residues based on the alignment of similar structures; it could potentially miss similarities in function among proteins that are split over different folds. Since the method uses sequence alignment, it could miss motifs or families with non-conserved sequence. Also, there is no mechanism to exclude profiles of one family from occurring in another family, which our method achieves by enforcing a maximum background occurrence for each motif.

ProKnow [99] correlates sequence and global/local structural features found in a new protein with an extensive function knowledge-base, weighting predictions by the Bayes theorem. ProKnow's impressive accuracy depends on the accuracy of GO annotations for existing proteins; unreliable annotations marked "Inferred from Electronic Annotation" are excluded, since they decrease function prediction accuracy from 89% to 56%. ProKnow is most accurate at detecting unspecific top-level functions (e.g. protein binding), whereas our method excels at precise identification of a specific functional family, whose motifs differentiate it from closely related families. The RIGOR method [100] included in ProKnow detects graph patterns from known functional sites; if extended to detect family-specific motifs, it could find uncharacterized protein families, ensure robustness and family-specificity in local structures matches. Thus, family-specific motifs that are identified with our method complement [99], ProFunc [65] and other meta servers for protein function inference.

Limitations

Our method has limitations, arising from representation choices, algorithmic issues, and the nature of the problem itself. In our representation, we use C_α coordinates to calculate graph edges and their lengths; this choice captures shared topology, but may miss contacts made by long side-chains. Currently we do not allow residue substitutions in patterns, other than unifying V,A,I,L. Merging commonly substituted residue types (e.g. D and E) increases the sensitivity of motifs but must decrease their specificity, losing motifs that are no longer unique to a family. Finally, the weighted edge matching criteria may be too restrictive to find patterns with widely varying geometry or containing edges that happen to lie on bin boundaries. We have developed a new overlapping-bin weighted edge representation to remedy this last problem [79].

Algorithmically, subgraph mining involves the NP-complete problem of subgraph isomorphism. The FFSM algorithm [80] stores graph embeddings, so it does well with small isomorphic subgraphs, but can become inefficient with the large ones that could arise in families with very similar or identical structures. In these cases, however, it may be more appropriate to use global sequence or structure similarity methods for functional annotation in place of motif mining, since by design the latter approach is more applicable to proteins with remote sequence or structure homology.

It is part of the nature of the problem that classifications that are too fine can produce too many motifs due to high local similarity or small sample sizes, such as families with three or fewer members. Conversely, too coarse a classification can produce no motifs that are specific to a family—this happens with 35% of our SCOP families and superfamilies, especially the latter because of their heterogeneity. Because the number, specificity, and sensitivity of motifs depends on size and heterogeneity of the family, the support and background occurrence parameters must be varied to find meaningful sets of motifs for the maximum number of families.

Conclusions

We have described a fast and robust method for protein function prediction based on structure-based residue packing patterns identified as family-specific motifs. This paper presented novel algorithms for motif mining in SCOP or GO families as well as approaches for matching an orphan protein to a family based on the occurrence of family motifs in the orphan protein. The chief results reported in this paper could be summarized as follows. (1). We have processed all families in both SCOP and EC classifications (at different levels of hierarchy) and identified subsets of families where we could mine statistically significant family specific motifs. For each family, we have identified family specific motifs that collective amount to a unique Motif Library. (2). Based on family complexity and the results of motif mining we have developed a family nomenclature that classifies families into several groups: Ideal, Rich, Poor, Functionally heterogeneous, Tiny, Omitted, and Augmented. (3). For efficient mining of a query orphan protein against a Motif Library, we have developed a novel special "local neighborhood" graph index. (4). Special metrics of statistical significance of function inference based on motif matching have been introduced and discussed. These algorithmic and computational developments lay a foundation for the experimental application and validation of our approach. In the following companion paper (Part II, [10]) we describe the validation of the method, and discuss several case studies of function inference that are relevant to target identification for drug discovery.

Acknowledgments These studies were supported by NIH grant GM068665 and NSF grant CCF-0523875.

References

1. Overington J, Al-Lazikani B, Hopkins A (2006) Nat Rev Drug Discov 5:993
2. Holm L, Sander C (1996) Science 273:595
3. Smith LM (1989) Genome 31:929
4. Burley SK (2000) Nat Struct Biol 7 Suppl:932
5. Koonin EV, Galperin MY (2002) Sequence-evolution-function: computational approaches in comparative genomics. Kluwer Academic Publishers, Dordrecht, The Netherlands (published online on NCBI bookshelf, 2003)
6. Aloy P, Querol E, Aviles FX et al (2001) J Mol Biol 311:395
7. Bandyopadhyay D, Huan J, Liu J et al (2006) Protein Sci 15:1537
8. Huan J, Bandyopadhyay D, Wang W et al (2005) J Comput Biol 12:657
9. Huan J, Wang W, Prins J (2003) ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining
10. Bandyopadhyay D, Huan J, Prins J et al (2009) J Comput Aided Mol Des. doi:10.1007/s10822-009-9277-0
11. Gherardini P, Helmer-Citterich M (2008) Brief Funct Genomic Proteomic 7:291
12. Zhao X, Chen L, Aihara K (2008) Amino Acids 35:517
13. Redfern O, Dessailly B, Orengo C (2008) Curr Opin Struct Biol 18:394
14. Rost B (1999) Protein Eng 12:85
15. Tian W, Skolnick J (2003) J Mol Biol 333:863
16. Hofmann SK, Bucher P, Falquet L et al (1999) Nucleic Acids Res 27(1):215
17. Gribskov M, Luthy R, Eisenberg D (1990) Meth Enzymol 183:146
18. Altschul SF, Madden TL, Schaffer AA et al (1997) Nucleic Acids Res 25:3389
19. Krogh A, Brown M, Mian IS et al (1994) J Mol Biol 235:1501
20. Madera M, Gough J (2002) Nucleic Acids Res 30:4321
21. Lichtarge O, Bourne HR, Cohen FE (1996) J Mol Biol 257:342
22. Kristensen D, Ward R, Lisewski A et al (2008) BMC Bioinformatics 9:17
23. Ward R, Erdin S, Tran T et al (2008) PLoS ONE 3:e2136
24. Koonin EV, Makarova KS, Aravind L (2001) Annu Rev Microbiol 55:709
25. Tatusov RL, Koonin EV, Lipman DJ (1997) Science 278:631
26. Bowers PM, Pellegrini M, Thompson MJ et al (2004) Genome Biol 5:R35
27. Date SV, Marcotte EM (2005) Bioinformatics 21:2558
28. Thomas J, Ramakrishnan N, Bailey-Kellogg C (2008) IEEE/ACM Trans Comput Biol Bioinform 5:183
29. Song N, Joseph J, Davis G et al (2008) PLoS Comput Biol 4:e1000063
30. Lanczycki C, Chakrabarti S (2008) Bioinformation 2:279
31. Espadaler J, Eswar N, Querol E et al (2008) BMC Bioinformatics 9:249
32. Taylor W, Orengo C (1989) J Mol Biol 208:1
33. Andreeva A, Howorth D, Brenner SE et al (2004) Nucleic Acids Res 32:D226
34. Orengo C, Michie A, Jones S et al (1997) Structure 5:1093
35. Gibrat J, Madej T, Bryant S (1996) Curr Opin Struct Biol 6:377
36. Krissinel EB, Henrick K (2004) Softw Pract Exp 34:591
37. Holm L, Sander C (1997) In: Gaasterland T, Karp PD, Karplus K, Ouzonis CA, Sander C, Valencia A (eds) ISMB'97. 5th International conference on intelligent systems for molecular biology, Halkidiki, Greece, June 1997, p 140
38. Hegyi H, Gerstein M (1999) J Mol Biol 288:147
39. Glaser F, Pupko T, Paz I et al (2003) Bioinformatics 19:163
40. Liang M, Brutlag D, Altman R (2003) In: Altman RB, Dunker AK, Hunter L, Jung TA (eds) PSB'03. 8th Pacific symposium on biocomputing, Hawaii, January 2003, p 204
41. Russell RB (1998) J Mol Biol 279:1211
42. Stark A, Russell R (2003) Nucleic Acids Res 31:3341
43. Stark A, Shkumatov A, Russell RB (2004) Structure (Camb) 12:1405
44. Bradley P, Kim PS, Berger B (2002) Proc Natl Acad Sci 99:8500
45. Jambon M, Andrieu O, Combet C et al (2005) Bioinformatics 21:3929
46. Nussinov R, Wolfson HJ (1991) PNAS 88:10495
47. Barker J, Thornton J (2003) Bioinformatics 19:1644
48. Shulman-Peleg A, Nussinov R, Wolfson H (2004) J Mol Biol 339:607
49. Binkowski TA, Freeman P, Liang J (2004) Nucleic Acid Res 32:W555
50. Laskowski RA, Luscombe NM, Swindells MB et al (1996) Protein Sci 5:2438
51. Ferre F, Ausiello G, Zanzoni A et al (2004) Nucleic Acids Res 32:D240
52. Taylor WR, Jonassen I (2004) Proteins 56:222
53. Artymiuk PJ, Poirrette AR, Grindley HM et al (1994) J Mol Biol 243:327
54. Gardiner EJ, Artymiuk PJ, Willett P (1997) J Mol Graph Model 15:245
55. Samudrala R, Moult J (1998) J Mol Biol 279(1):287
56. Schmitt S, Kuhn D, Klebe G (2002) J Mol Biol 323(2):387
57. Stark A, Sunyaev S, Russell RB (1998) J Mol Biol 326:1307
58. Wangikar PP, Tendulkar AV, Ramya S et al (2003) J Mol Biol 326:955
59. Milik M, Szalma S, Olszewski K (2003) Protein Eng 16(8):543
60. Turcotte M, Muggleton S, Sternberg M (2001) J Mol Biol 306(3):591
61. Fetrow JS, Skolnick J (1998) J Mol Biol 281:949
62. Murga L, Wei Y, Ondrechen M (2007) Genome Inform 19:107
63. Xie L, Bourne P (2007) BMC Bioinformatics 8 Suppl 4:S9
64. Weskamp N, Kuhn D, Hullermeier E et al (2004) Bioinformatics 20:1522
65. Laskowski RA, Watson JD, Thornton JM (2005) Nucleic Acids Res 33:W89
66. Mulder N, Apweiler R (2008) Curr Protoc Bioinformatics Chapter 2: Unit 2.7
67. Gough J, Chothia C (2002) Nucleic Acids Res 30:268
68. Hendlich M, Bergner A, Gunther J et al (2003) J Mol Biol 326:607
69. Porter CT, Bartlett GJ, Thornton JM (2004) Nucleic Acids Res 32:D129
70. Jones S, Barker JA, Nobeli I et al (2003) Nucleic Acids Res 31:2811
71. Milner-White EJ, Nissink JW, Allen FH et al (2004) Acta Crystallogr D Biol Crystallogr 60:1935
72. Laskowski R, Watson J, Thornton J (2005) J Mol Biol 351:614
73. Watson J, Sanderson S, Ezersky A et al (2007) J Mol Biol 367:1511
74. Bandyopadhyay D, Snoeyink J (2004) ACM-SIAM Symposium On Discrete Algorithms. New Orleans, LA, USA
75. Ullman JR (1976) J Assoc Comput Mach 23:31
76. Bairoch A (2000) Nucleic Acids Res 28:304
77. Gene Ontology Consortium (2004) Nucleic Acids Res 32:D258

78. Wang G, Dunbrack RL (2003) Bioinformatics 19:1589 http://www.fccc.edu/research/labs/dunbrack/pisces/culledpdb.html
79. Huan J, Bandyopadhyay D, Snoeyink J et al (2006) IEEE Computational Systems Bioinformatics Conference (CSB). Stanford, CA, USA
80. Huan J, Wang W, Bandyopadhyay D et al (2004) In: Gusfield D, Bourne P, Istrail S (eds) RECOMB'04. 8th Annual international conference on research in computational molecular biology, San Diego, April 2004, p 308
81. Huan J, Wang W, Prins J et al (2004) In: Kohavi R, Gehrke J, DuMouchel W, Ghosh J (eds) ACM SIGKDD'04. 10th International conference on knowledge discovery and data mining, Chicago, August 2004, p 581
82. Pegg SC, Brown S, Ojha S et al (2005) In: Altman RB, Dunker AK, Hunter L, Jung TA (eds) PSB'05. 10th Pacific symposium on biocomputing, Hawaii, January 2005, p 358
83. Babbitt PC (2003) Curr Opin Chem Biol 7:230
84. Wilson CA, Kreychman J, Gerstein M (2000) J Mol Biol 297:233
85. Lindqvist Y, Schneider G (1997) Curr Opin Struct Biol 7:422
86. Grishin NV (2001) J Struct Biol 134:167
87. Keller J, Smith P, Benach J et al (2002) Structure 10:1475
88. Fetrow JS, Siew N, Di Gennaro JA et al (2001) Protein Sci 10:1005
89. Michalovich D, Overington J, Fagan R (2002) Curr Opin Pharmacol 2:574
90. Hegyi H, Gerstein M (2001) Genome Res 11:1632
91. Nagano N, Orengo C, Thornton J (2002) J Mol Biol 321:741
92. Petsko G, Ringe D (2004) Protein structure and function. New Science Press Ltd, Waltham, MA, USA
93. Leibowitz N, Fligelman Z, Nussinov R et al (2001) Proteins 43:235
94. Wang K, Samudrala R (2006) BMC Bioinformatics 7:278
95. Hambly K, Danzer J, Muskal S et al (2006) Mol Divers 10:273
96. Xie L (2004) WIPO patent http://www.wipo.int/pctdb/en/wo.jsp?WO=2005045424
97. Xie L, Bourne P (2008) Proc Natl Acad Sci USA 105:5441
98. Pazos F, Sternberg MJ (2004) Proc Natl Acad Sci USA 101:14754
99. Pal D, Eisenberg D (2005) Structure (Camb) 13:121
100. Kleywegt GJ (1999) J Mol Biol 285(4):1887