A new library of surface patches - design and applications

Roi Gamliel[1], Klara Kedem[1], Rachel Kolodny[2], and Chen Keasar[1]*

[1]Computer Science Department, Ben-Gurion University

[2]Computer Science Department, Haifa University

Technical report TR 09-10, Computer Science Department, Ben-Gurion University

chen@cs.bgu.ac.il

Abstract

Protein surfaces serve as an interface with the molecular environment and are thus tightly bound to proteins' function. On the surface, geometric and chemical complementarity to other molecules provides interaction specificity for ligand binding, docking of bio-macromolecules, and enzymatic catalysis.

As of today, there is no accepted general scheme to represent protein surfaces. Furthermore, most of the research on protein surface focuses on regions of specific interest such as interaction, ligand binding and docking sites. We present here a more general approach for surface representation, as we wish to study the entire surface regardless of its functional roles.

In this work we characterize protein surfaces using *surface patches,* which are small fractions of the surface of proteins. We define a heuristic distance measure between surface patches which is used to cluster the patches from a large set of non redundant proteins. The resulting set of cluster centers (*centroids*) provides representative surface patch structures and constitutes our *surface patch library*.

To evaluate the biological significance of our method we examined the ability of the library to capture surface characteristics of native protein structures as opposed to those of decoy sets generated by state of the art prediction methods. To this end we compared the compatibility of native proteins and decoys to the library, and attempted to rank protein structure predictions. We found that the patches of the decoys were less compatible with the library than their corresponding native structures. Thus, we were able to use the patches to distinguish native models from models generated by servers. This trend, apparently, does not extend further to the decoys themselves as we failed to reliably rank decoys by their surface patches characteristics.

We expect that this high-quality, generic surface patch library will add a new perspective to the description of protein structures and will improve our ability to predict them. Particularly, we expect that it will help improve the prediction of surface features that are apparently rather neglected by current techniques.

1. Introduction

Protein surfaces serve as interfaces with the molecular environment and are thus tightly bound to proteins' function. On the surfaces, geometric and chemical complementarity to other molecules provides interaction specificity for ligand binding, docking of bio-macromolecules, and enzymatic catalysis. Further, catalysis itself is a surface phenomenon. Thus, surface analysis enables the discovery of functional relationships between proteins that are only distantly related or even unrelated evolutionarily. In their pioneering work Fischer et al. *[36]* were able to identify the active site residues of subtilisin and sulphydryl proteases by querying a large dataset with the catalytic triad of the (apparently) evolutionary unrelated trypsin. That work was later followed by similar ones that studied more subtle features of functional surface regions. Lin et al. *[44]* represented molecular surface as a limited number of critical points disposed at key locations over the surface. Using this representation, they were able to achieve accurate protein-protein and protein-small molecule docking. Norel et al. *[45]* adapted the geometric hashing paradigm, originally developed in the context of computer vision, for the docking problem, and by using an indexing approach based on a transformation invariant representation, they scanned groups of surface dots (or atoms) and detected optimally matched surfaces. Pickering at al. *[15]* matched functional features on protein surfaces thus showing functional relationships (NAD binding) between proteins where sequence alignment and fold comparison failed. Wolfson et al. *[32]* and Zhou et al. *[43]* found that shared patterns on protein surfaces reveal similarities between RNA dinucleotide binding sites of proteins with different overall sequences, folds and functions. Finally, Binkowski & Joachimiak *[37]* were able to identify Hem binding pockets in a variety of evolutionary unrelated proteins. A major characteristic of all these studies is that they took a supervised approach. They first identified and learnt a functional region within a dataset of proteins and then used the acquired knowledge to annotate other, evolutionary unrelated proteins.

A prerequisite for the study of protein surfaces is a method for their identification and representation. Much work has been devoted to this issue over the years and several approaches have been suggested to address it. We will explore some of the developments in this area, but for a complete review of the subject we refer the reader to Via et al. *[14]*. The first to define the solvent accessible surfaces of proteins were Lee and Richards *[25]* in 1971. They developed a method to identify surface atoms and to differentiate them

from buried ones. The union of the surface atoms defined the accessible surface. In 1983 Connolly *[27]* introduced the first exact analytical method for computing the accessible surface area that smooth out the molecular surface by eliminating narrow cavities that cannot be reached by a spherical probe (typically having 1.4Å radius, like a water molecule). Over the years this method has been improved, especially in terms of efficiency *[34][35]* and as for today it is the most widely used approach. New developments in computational geometry, in the field of alpha shape theory by Liang and Edelsbrunner *[33]*, provide an accurate method to describe the topological structure of a molecule and to efficiently compute the solvent accessible surface.

The complexity of these fine grained representations of the protein surface calls for coarse graining by some kind of abstraction of the intuitive concept of surface patches. In their pioneering work Jones and Thornton *[5][6]* used surface patches to identify potential sites for protein-protein interactions. They defined surface patches as overlapping, roughly circular, sets of proximate surface residues (the size of which depends on the type of the target interaction), and compared binding site patches with non-binding ones. The unique binding site features were used to derive an interaction propensity score for each patch, which was later implemented in a web based bioinformatics tool to predict potential protein-protein interaction sites *[7]*. A decade later Baldacci et al. *[2][3][4]* used a very different approach to surface patches to tackle a very different task, that of identifying structural similarity and plausible evolutionary connection between proteins. Their method aims to represent protein surfaces as graphs and thus, make them amenable for study using graph theoretic algorithms. To this end they determined, for each protein, a collection of variable size, non-overlapping surface patches, each of which included a set of homogenous and connected surface points. The patches were then classified to one of 12 predetermined types. Patches that were not compatible with any of these types ("gray" patches) were ignored. This strict reduction in complexity allowed the authors to represent protein surfaces by graphs whose nodes are patches and the edges are labeled with inter-patch distances. They found frequent sub-graphs of patterns of patches through data mining techniques and used these sub-graphs to classify the proteins. While very different, both applications are tightly tailored towards their specific aims and it is hard to see how they can be used in a different context.

The high dependence of the above realizations of surface patches on the context in which they are applied is in sharp contrast to another common implementation of protein structure concept, namely *fragments* (i.e., continuous structural stretches along the protein chain). Since the introduction of fragments some twenty years ago by Jones et al. *[8]* they were used for a wide range of applications and had a profound impact on protein structural biology. The original work of Jones et al. *[8]* used the fragments for the interpretation of electron density maps. Later they were used to study sequence structure relationships by Unger et al. *[9],* Han & Baker *[38]* and Kolodny et al. *[1]*. Moreover, they were used for sequence alignment by Ye et al. *[39],* protein structure prediction by Levitt et al. *[41],* protein structure comparison and classification by Koehl et al. *[42,]* and for large scale mapping of the whole fold space of proteins by Friedberg and Godzik *[40]*. The cornerstone of all these studies is the use of clustering techniques to extract a limited number of representatives from the vast dataset of known protein fragments.

The major motivation of our study is our belief that a similar approach may also benefit the study of protein surfaces and that a wide variety of applications may arise from a generic representation of surface patches. Accordingly, the current study presents a new approach to the identification of surface patches in proteins. We aim to study protein surfaces in an unsupervised way by building an unbiased library of representative surface patches.

To obtain such a library we find for each β-carbon (*pivot*), which lays on the protein surface, a group of surface atoms that surround it within a certain radius. Thus, each pivot defines a surface patch (Figure 1.a) and neighboring patches may overlap. We developed a heuristic distance measure between surface patches and used it to cluster a large number of patches extracted from a non-redundant dataset of protein structures (Figure 1.b). Each cluster consists of similar patches and is represented by its *centroid* which is the patch that minimizes the intra-cluster distance. These centroids serve as representative surface patch structures (Figure 1.c) and constitute our surface patch library. To evaluate the biological significance of our method we examined the ability of the library to capture surface characteristics of native protein structures as opposed to those of decoy sets generated by state of the art prediction methods. To this end we examined the fit of native proteins and decoys to the library, tried to rank protein structure predictions of five leading servers taken from CASP8, and analyzed the composition of two patches in an exemplary cluster.

The rest of the paper is structured as follows. In Section 2 we define surface patches, the clustering framework used, data sets and the distance function. Section 3 reports the tests we carried out and their significance. Finally, in Section 4 we discuss the significance of these results and suggest future studies.

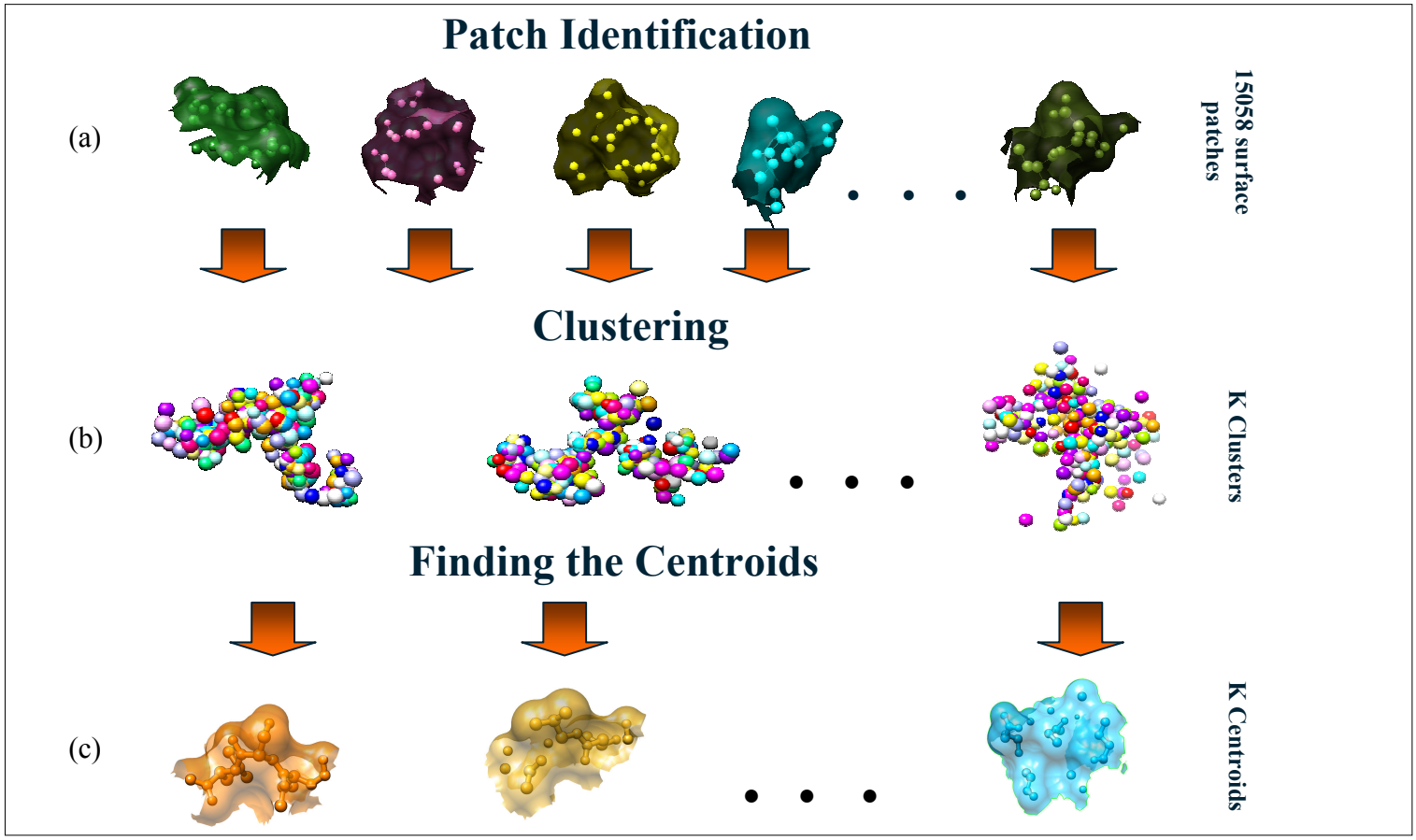

Figure 1: An overview of the construction of the surface patch library. (a) – The surface patches extracted from the data set, the atoms are shown as small spheres in the patch. (b) – Grouping the patches into k clusters. The atoms of the patches in each cluster are superimposed on the cenroid. For clarity, we omitted the surfaces. (c) – The cluster centroids constitute the surface patch library.

2. Methods

2.1 Parameter assignment

The methods outlined below require the assignment of quite a few free parameters that affect the results, sometimes in a rather unpredictable way (Table 4). Unfortunately, no rigorous theory exists regarding the choice of these parameters and an exhaustive search over all parameters space in unfeasible. Instead we had to do with a sparse local sampling of a single parameter at a time. In order to keep this document concise we have chosen not to elaborate on this parameter sampling process. Instead, the tested values as well as the current ones, which were used to produce the reported results, are summarized in Table 4 and we refer to this table whenever one of these parameters is mentioned.

It should be stressed though that in all the experiments, with all parameter values tested we got qualitatively similar results and specifically statistically significant result.

2.2 Data Sets

Our training set (Table 5), which is identical to the one used by Kolodny et al. *[1]*, includes 200 unique and high quality domains from SCOP version 1.57 *[19]*. Specifically, each of these domains had the highest ranking SPACI scores *[20]* in its SCOP category, where the SPACI score is a measure of the reliability and precision of a crystallographically-determined structure in a PDB file. The structures were retrieved from the Protein Data Bank (PDB) *[18]*.

The test set (Table 6) includes CASP8 single domain targets whose structures were solved by crystallography, and their respective models generated by the 5 best CASP8 servers (a total of 6 structures per target). Moreover, in the case of two servers (Baker and Zhang) we examined all the five models that they submitted for each of the above mentioned targets (additional 4*2 models per target). In order to ensure the independence of the training set and the test set we ran BLAST *[48]* search of the CASP8 targets against the training set. Targets with either an E-value below 0.0001 or more than 30% identity with any of the training set proteins were removed. After the removal of training set homologs the test set included 66 proteins.

2.3 Surface identification

The *accessible surface area* of each atom (measured in $Å^2$) was computed using PROGEOM *[12]*. For the binary distinction between surface and buried atoms we introduced a threshold of surface area value for each atom type (e.g., alanine-Cα or lysine-N_ζ). This threshold is based on the cumulative distribution of accessible surface area over all atoms of a specific type (Figure 2). We defined ***surf*** to be the area corresponding to the 99th percentile of the distribution and considered it the surface of a fully exposed atom. The highest values obtained were very noisy and we suspected that they may correspond to errors in the structures (e.g. missing side-chains that superficially expose backbone atoms). The threshold was defined as a fraction, ***surf****(1-***α***), of this value. The larger ***α***, the less atoms are considered to be on the surface. Using small ***α*** values added atoms in cavities to the surface. Using a high ***α*** value resulted in a very scattered coverage of the surface. This work uses ***α***=10% (Table 4).

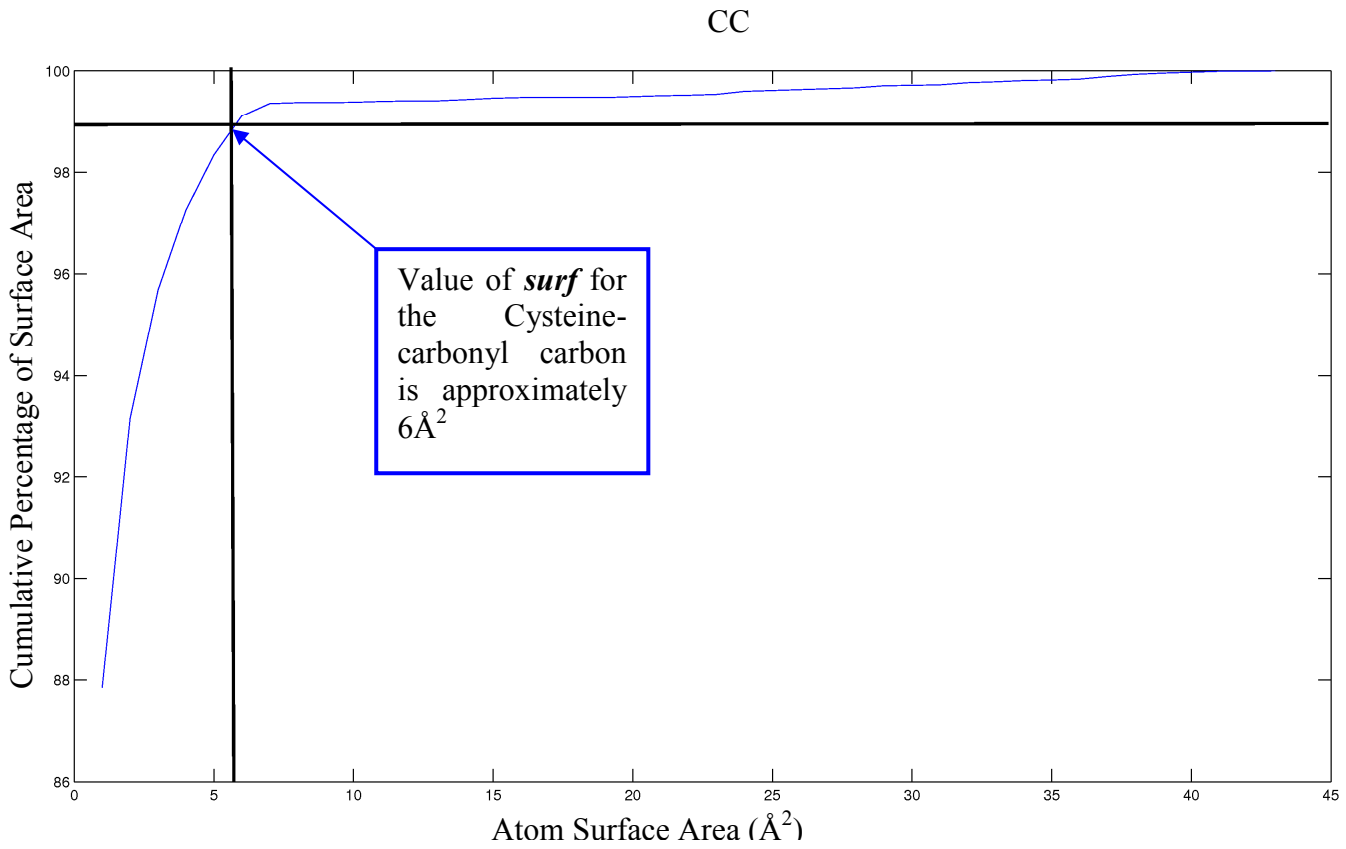

Figure 2: The estimation of ***surf*** - the maximal surface area of atoms of a given type. The cumulative surface area distribution of all the cysteine-carbonyl carbon atoms in the training set is plotted. We set ***surf*** to be the area in the 99th percentile of the

2.4 Patch definition

A patch is defined by a central surface β-carbon, which we call *pivot*, and a radius ***R*** (see table 4). It includes the surface atoms that are at distance not greater than ***R***Å from the pivot. Under this definition neighboring patches on the protein surface typically overlap (Figure 3).

The reasoning behind this somewhat arbitrary definition is that most β-carbons of surface residues are exposed, thus allowing a packed and uniform coverage of the surface. A different selection, such as α-carbon would have resulted in a scattered coverage of the surface, as the α-carbons are frequently buried. The patch radius ***R*** determines the patch's size, area of coverage and number of atoms. This work uses ***R***=7Å. Future work may incorporate several libraries, where each library is assembled for a different radius.

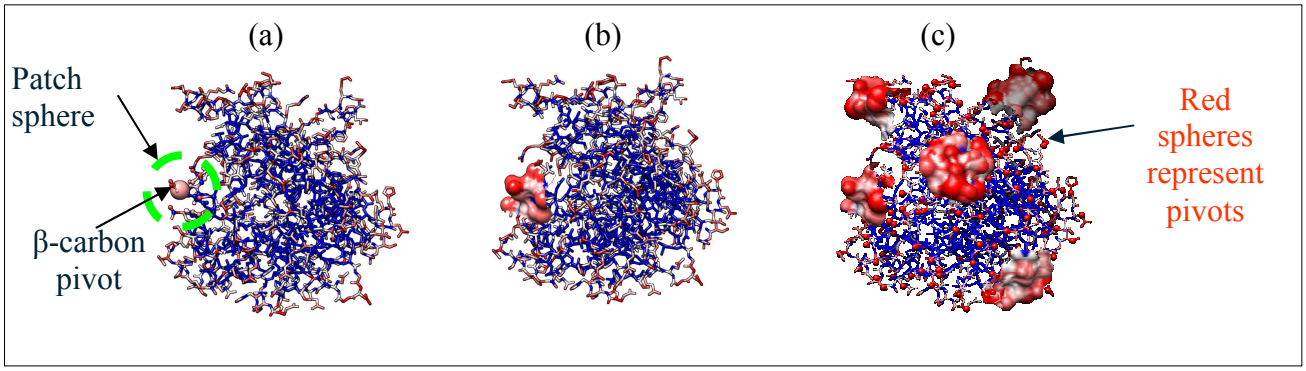

Figure 3: Surface patch definition (PDB code 12as [58]) as an example. Atoms are rendered by their surface exposure (red = exposed, blue = buried). (a) A patch is defined by a central surface β-carbon and a sphere around it (green). (b) The surface of a single patch. (c) Neighboring patches on the protein surface typically overlap (red spheres represent pivots).

2.5 Definition of distance between patches

Given two patches A and B such that A = $\{a_1,\ldots,a_n\}$ and B = $\{b_1,\ldots,b_m\}$, m ≤ n, we look for their optimal superposition both in terms of structure and of chemical properties. The distance value is the minimal root mean square deviation (RMSD) between the patches under a set of chemical constraints. If the compositions (see below) of the patches are too remote to allow meaningful superposition the distance is taken to be infinity. Formally:

1. Let T = $\{T_1,\ldots,T_{Ntypes}\}$ be the set of atom types.
2. Let ***t:*** {set of all atoms} → T be a mapping so that for an atom ***a***, ***t***(***a***) is the atom's type.
3. Let A = $\{a_1,\ldots,a_n\}$ and B = $\{b_1,\ldots,b_m\}$ be two sets of atoms to be superimposed.
4. Let the composition of patch A be $T_A = \{T_{1A},\ldots,T_{NtypesA}\}$ the vector of integers such that $\sum T_{iA} = n$ and, for each i, T_{iA} is the number of atoms of type T_i in set A.
5. The composition of patch B is similarly defined.
6. A and B are compatible if $|n-m| < \Phi_1$ or if $\exists_i |T_{iA} - T_{iB}| < \Phi_2$ or if $|rg(A) - rg(B)| < \Phi_3$ where *rg(A)* and *rg(B)* are the radii of gyration of sets A and B respectively and Φ_1, Φ_2, and Φ_3 are threshold values for size difference, chemical difference, and radius of gyration difference, respectively.
7. Let F = $\{f_1,\ldots, f_k\}$ be the set of all proper mappings of A and B. A mapping is a bipartite graph whose disjoint sets are A and B. A proper mapping ***f*** satisfies ***f***(***a***) = **b** iff ***f***(***b***) = **a** and ***t***(***a***) = ***t***(***b***), i.e. a pairing is made between atoms of the same type.
8. Then, the distance between A and B is:

$$D(A,B) = \begin{cases} \min_{f \in F} RMS(A,B,f) & if\ A\ and\ B\ are\ compatible \\ \infty & otherwise \end{cases}$$

Where *RMS(A,B,f)* is the optimal superposition of the atoms of A and B that are mapped by f. In practice, finding the optimal mapping is a hard combinatorial optimization problem, although the requirement for compatibility provides a filter that reduces the number of these calculations considerably. Thus, the use of the exact distance definition above might have rendered the calculation of numerous distances infeasible. Instead, we use a heuristic approximation that reduces the number of tested mappings.

To this end we define the inner sphere of a patch to be a sphere of radius ***r*** (see Table 4), ***r*** < ***R***, centered at the pivot β-carbon (Figure 4.a). We then exhaustively enumerate all possible chemically valid mappings between the inner sphere of one patch and the inner sphere of the other patch (Figure 4.b). The RMSD is measured after optimal least-squares superposition *[21]*. If the RMSD between the inner spheres is below a given threshold, it serves as a seed to match the full patches A and B. If no seed was found, the distance between the patches is taken to be infinity. Otherwise, the transformation coupled with each RMSD that passed the threshold is applied to the full patches and each atom of A is matched according to proximity and chemical attributes to the best fitting atom in B. Now we have a matching between A and B for each seed. For each such matching we compute the RMSD between A and B and pick the matching that yields the lowest RMSD.

This approximation is sensitive to the radius of the inner sphere. When the inner sphere has many atoms the runtime is very high. When, on the other hand, the inner sphere is almost empty we do not have enough points to compute the RMSD. Thus, we start each calculation with ***r***=4Å and if the number of atoms within the sphere gets below 4 or exceeds 9 we resize the radius of the inner sphere by one Ångström.

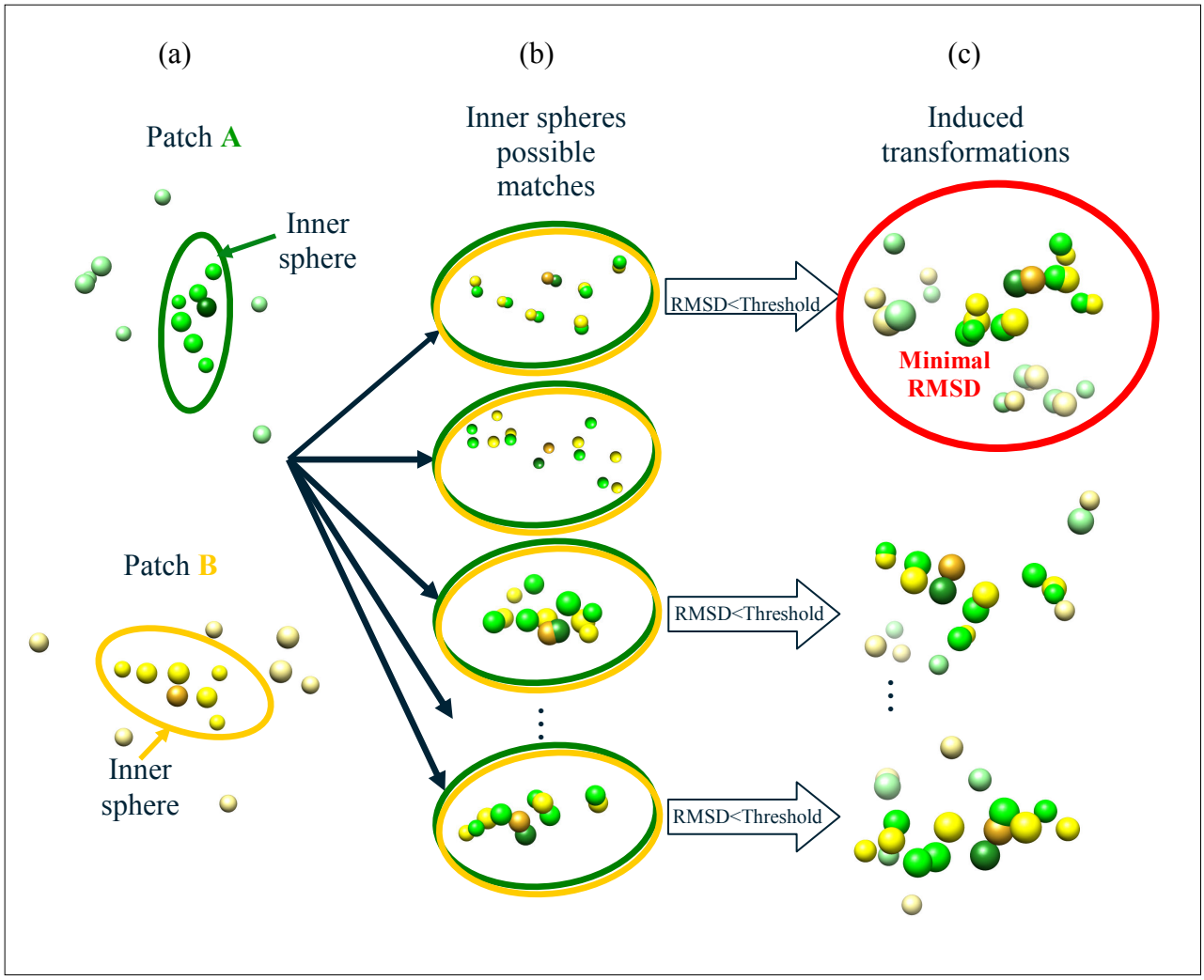

Figure 4: The distance approximation. (a) Patches A and B; Inner sphere circled (the pivots for patch A and B are colored dark green and yellow respectively). (b) Enumerating the inner spheres' possible matches. (c) If the RMSD between the inner spheres is below a certain threshold, it acts as a seed to mapping of the full patches. Finally we pick the superposition that yields the minimal RMSD value (encircled in red).

2.6 Outlier weeding

Outlier patches are distant from the majority of all other patches. While some of them may represent unique structural features (e.g., binding site of some rare type) others are likely to be the result of problems in the crystallographic data (e.g. missing side-chains). Distinguishing between the two categories might have required considerable manual work. On the other hand, if ignored, these outliers may dominate the clustering process and result in numerous non-informative singleton clusters. Thus, we have decided to weed

them out as a preprocessing step before the clustering. Formally outliers were defined as those patches that have a distance greater that 2.5 $Å^2$ from more than ***outlier_threshold***% (see Table 4) of the other patches. In this work ***outlier_threshold*** was set to 90% leaving out approximately 1.51% of the surface patches.

2.7 Patch clustering

For patch clustering we used the unsupervised K-means method *[22]*, which turned out useful in the related study of fragment clustering *[1]*. In a nutshell, this method partitions the patches into a pre-defined number, ***K*** (Table 4), of clusters while reducing the average distance of patches from their *cluster centroid,* where the cluster centroid (or center) is defined as the patch with the lowest sum of distances to the other cluster members. Thus, the centroids may be considered representatives of their respective clusters and the set of centroids as a representative of the whole patch set, from which the clusters were derived.

The k-means algorithm uses an iterative heuristic approach for the minimization of its cost function, sum of squared error (SSE), where the "error" is the distance between each patch and the centroid of its cluster (Figure 5). Each iteration is seeded by a set of ***K*** putative centroids and the clusters are formed by associating each patch with the closest putative centroid. Then, for each cluster we find the true centroid (typically not the one used as a seed) and the set of centroids is used to seed the next iteration. This process is repeated until convergence.

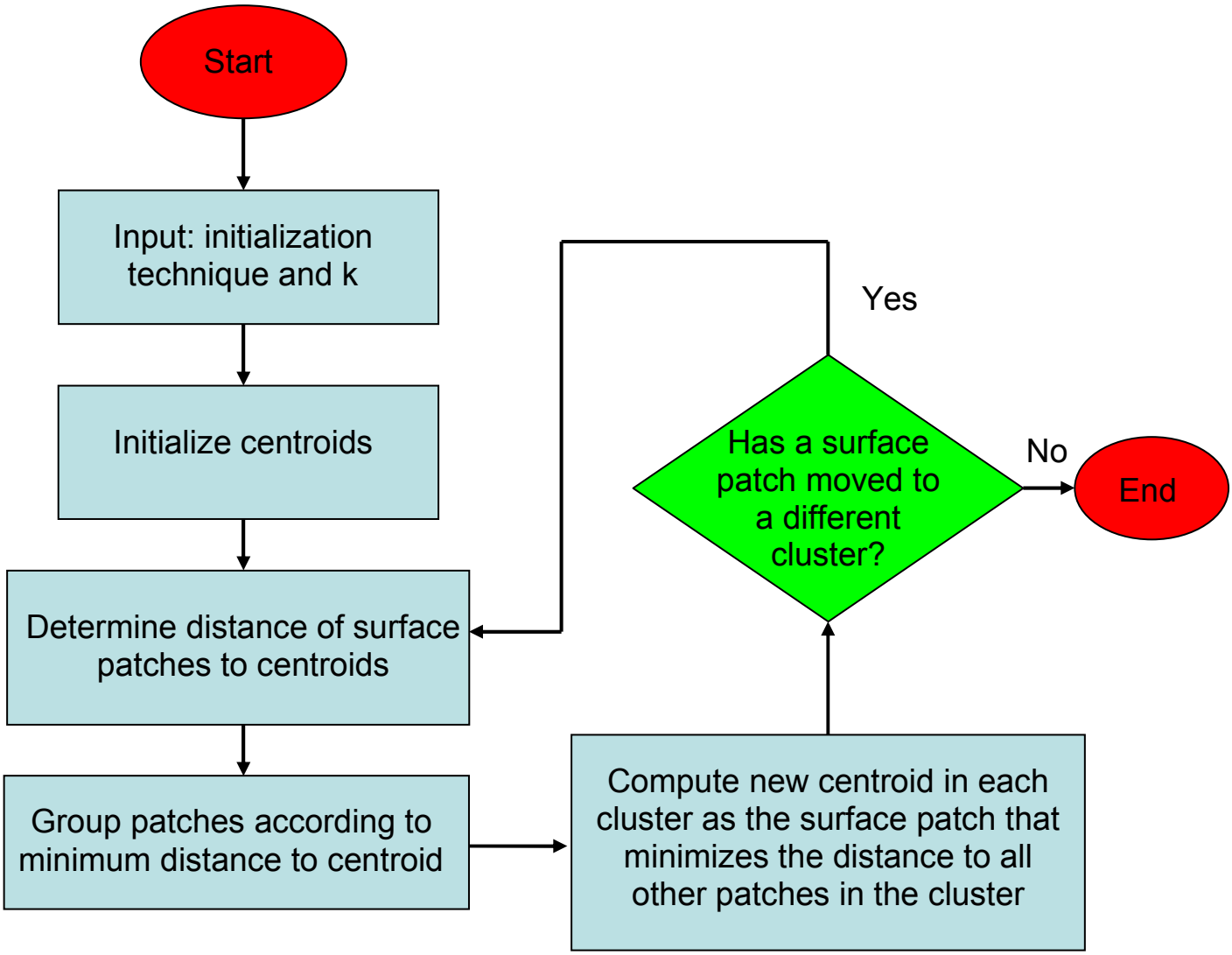

Figure 5: a flow chart for the k-means algorithm. The algorithm requires as input the initialization technique and the number of clusters ***K***. It then iterates until convergence, determining in each step the grouping of the patches according to minimum distance, until no movement of patches between clusters is made.

This algorithm is not optimal and thus its results depend on the initial assignment of candidate centroids. We have tested three methods to seed the first iteration:

- Random initialization
- Furthest first method *[23]*
- K-means ++ *[24]*

While all three methods produced meaningful results, K-means++ initialization achieved better results both in terms of SSE and in the applicative tests (see below). This method uses a randomized seeding

technique which is $\Theta(\log k)$-competitive with the optimal clustering, thus producing better clusters. In the K-means ++ initialization method we aim to spread ***K*** initial clusters away from each other. Here is a detailed description of the method:

- A surface patch is chosen from all data patches as a cluster center randomly with a uniform distribution.
- For each data surface patch x, compute $D(x)$, the distance between x and its nearest center among the centers already picked.
- Choose a new surface patch as center with probability proportional to $D(x)^2$.
- Repeat the above two steps until ***K*** cluster centers are chosen.

While the above-mentioned procedure requires extra time compared with the other initialization procedures, the K-means ++ algorithm converges faster and yields better results in respect to the cost function.

The decision about the proper value of ***K***, the number of clusters, is a somewhat delicate issue. A few large clusters may include almost unrelated members and thus, be non-informative. On the other hand, numerous small clusters may represent over training of the dataset and again have low predictive power. Several heuristic approaches to the assignment of ***K*** were suggested in the literature *[50]-[54]*. We tested several values finally deciding to use 350 as the number of clusters, since it produced the best results in the tests that were later carried out to evaluate the library (see section 3).

2.8 Statistical analysis

Bootstrap resampling

Bootstrap resampling *[46]* is a statistical method that allows an estimation of distribution properties from a small sample. One may create numerous samples from the single available sample by means of resampling with replacement. That is, the elements are sampled randomly with equal probability and each element choice is independent of the previous ones so that an element may be chosen more than once. The process of resampling from the original sample is done a number of times (as many times as the computing resources allow). This way, any property of the original sample is replaced by its distribution in the set of samples, which makes it more amenable for statistical inference.

Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov Test *[47]* is a non parametric statistical test for the significance of the difference between two distributions. Let X_1 and X_2 be two samples. The null hypothesis of the test is that X_1 and X_2 were sampled from the same underlying distribution. The alternative hypothesis is that they were sampled from different distributions. The test statistic is $max(|F_1(x)-F_2(x)|)$ where $F_1(x)$ is the proportion of X_1 values not greater than x and $F_2(x)$ is the proportion of X_2 values not greater than x. We used this test to evaluate the significance of difference between the distance distribution of native and decoy patches to our library. We used the 5% significance level for this test.

3. Results

A total of 15,288 surface patches were extracted from the training set domains and 15058 remained after outlier weeding. An all against all distance matrix was calculated and served as an input to the K-means++ algorithm, resulting in a library of 350 cluster centroids.

This section demonstrates the ability of this library to capture genuine aspects of native protein surfaces. We show significant difference between the patches of decoys and native structures. Furthermore, the library is useful in sorting out native protein structure from server structure predictions receiving statistically significant results.

3.1 The distribution of native and decoy patch distances from cluster centroids

First, we compared the distances between native protein patches to their respective closest cluster centroids with the corresponding distances of patches extracted from decoy models (Figure 6). The distance distribution of the native patches was the same as that of our training set. We found that decoy patches had significantly different distribution with a larger median distance from the centroids ($p<10^{-28}$).

3.2 Distinguishing native structures from decoy structures

We discovered that the average patch distance from the closest cluster representative was very meaningful in distinguishing native structures from a set of decoy structures (Table 1). We examined 66 CASP8 targets, inspecting the native structures and their respective model structures generated by 5 servers (model #1 of each server). For each of the above-mentioned structures, we examined its fit to the library. In 74% of the cases, out of the six structures examined for each target, the native structure was ranked as the one with best fit to our patch library, while the random expectation is 1/6, i.e., 16.6(±7.7) % (where the standard deviation of 7.7 was estimated by 10000 bootstrap re-sampling iterations).

3.3 Predicting the quality of a model

The above-mentioned results encouraged us to try and use the average patch distance from the closest cluster representative, as a way to predict the quality of a model, in terms of GDT and RMS. Since the results were similar we will only show them for the RMS measurement. We inspected models generated by the 5 servers for the 66 CASP8 targets. In 29% of the targets, the model which best fitted our library was indeed the model receiving the best score in CASP8, in terms of RMS, as can be seen in Table 2. Since the

random expectation is 20 (±7.5)% (where standard deviation was estimated by 10000 bootstrap resampling iterations) the meaning of this result is probably miniscule.

3.4 The relative size of clusters

Another interesting measure is the distribution of relative cluster sizes (ΔS, defined below). Let C be one of the library's clusters and Q a set of patches of one data set (e.g., the patches of Zhang's decoy set). Let p be a set of patches from Q that are associated with cluster C, i.e. patches that are closest to C's centroid. Then we define the *size* of C with respect to Q as $S(C,Q)=\frac{|p|}{|Q|}$ and the *relative size* (with respect to the training set) as $\Delta S(C,Q)=\left|S(C,Q)-S(C,\{TRAINING\_SET\_PATCHES\})\right|$.

Figure 7 compares the distributions of relative cluster sizes of the CASP8 native structures and the corresponding relative cluster sizes of the decoys from the various servers. Curiously, not only the native structures differ significantly from than the decoys ($p<10^{-3}$ on the average), the decoy structures differ among themselves with the Baker's server decoy group significantly further than the native compared with the other servers ($p<10^{-7}$).

3.5 Predicting the quality of a model in a specific server

Our results demonstrated an unpredicted ability to distinguish between different servers, as appears in Figure 7. Thereupon, we tried to predict the quality of a model in a specific server, again using the average patch distance from the closest cluster representative. We inspected 66 CASP8 targets, and their respective models generated by two servers, the Baker et al. server *[36]* and the Zhang server *[49]*. Each of these servers generated 5 models for each target, and in 28% and 23% of the targets we were able to predict the model that had been ranked best for the server, in terms of RMS (see Table 3) while the random expectation is 20%(±7.6) for Baker's server and 20%(±6.7) for Zhang's server (standard deviation estimated by 10000 bootstrap re-sampling iterations). Thus, apparently we are not yet able to use our library for this purpose.

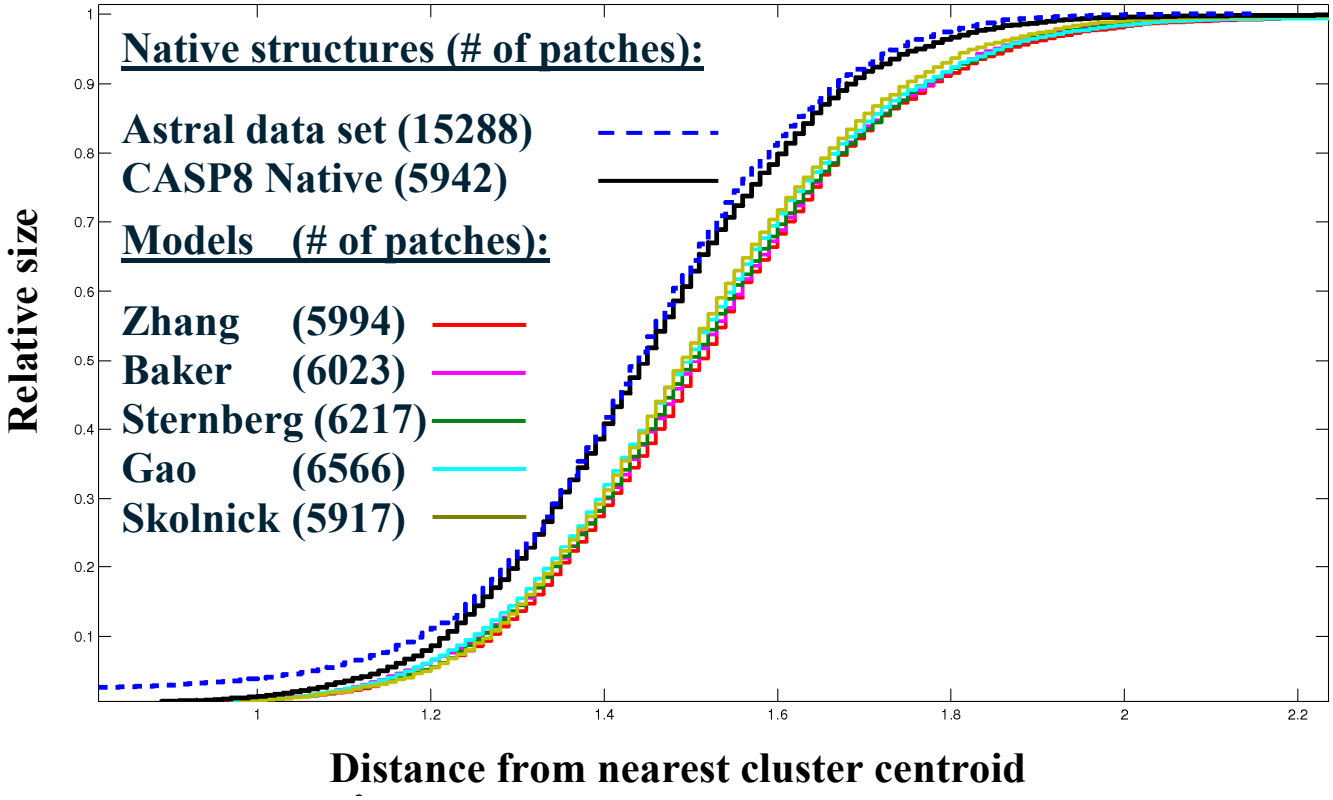

Figure 6: The distance of native patches and decoy patches to their closest cluster centroid in the patch library represented by cumulative distributions. As seen, native structures act significantly different from CASP8 server models in terms of the fit to our library.

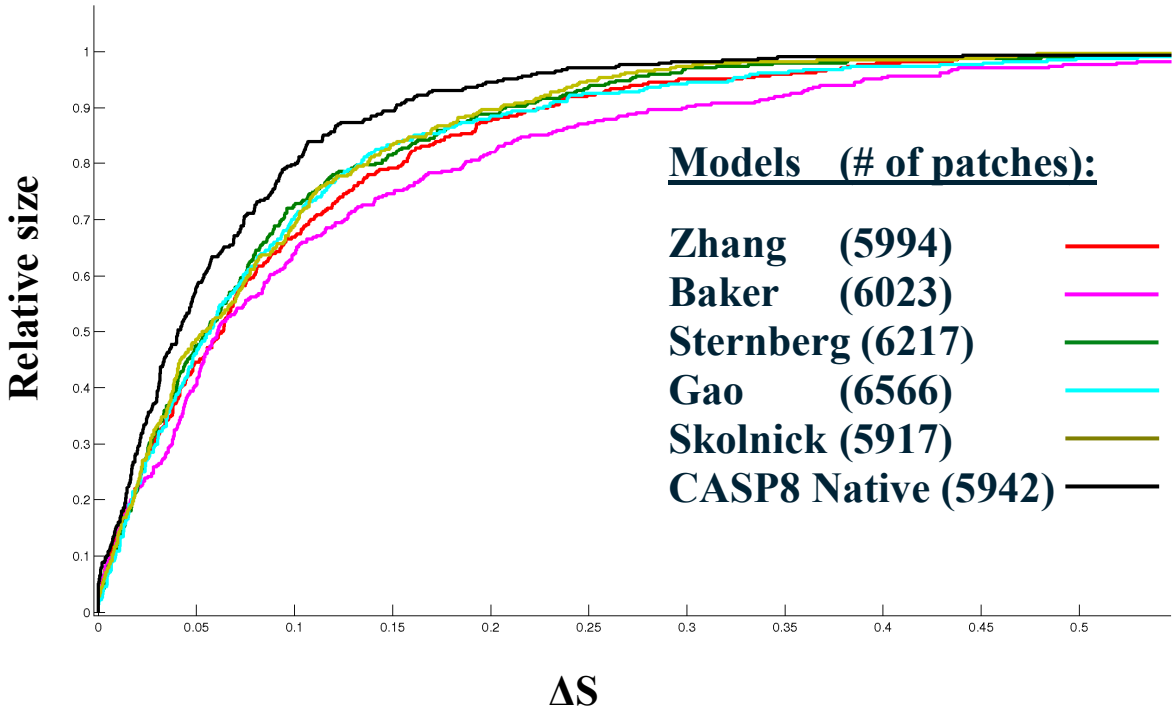

Figure 7: For each of the training set clusters, we computed their relative size with respect to the total number of patches of the training set and compared it to the corresponding values in the test set. ΔS is the absolute difference between the relative size of the native training set and its corresponding value in one of the test sets. The cumulative distribution for the ΔS values for each of the test sets is shown. Results show that the native training set is more similar in relative cluster size distribution to the native set than to the server test sets.

Rank	% rank and s.t.d
#1	74±7
#2	14±7
#3	3±7
#4	4.5±7
#5	4.5±7
#6	0±7

Table 1: Ranking the native structures among 6 conformations (of the native and the 5 servers' predictions). As seen, in 74% of the cases the native structure was ranked as the best fitting structure to the library. The random expectation is 16.6%.

Rank	% rank and s.t.d
#1	29±7
#2	20±7
#3	13±7
#4	20±7
#5	18±7

Table 2: Ranking the model with best RMS score among the 5 models generated by each of the 5 servers. We see that in 29% of the cases, the model that best fitted our patch library was indeed the model that got the best RMS score for its server. Compare to the random expectation which is 20%.

Rank	% rank Baker and s.t.d	% rank Zhang and s.t.d
#1	28±7	23±6
#2	17±7	25±6
#3	25±7	15±6
#4	7±7	14±6
#5	23±7	23±6

Table 3: Ranking the model with highest RMS score among 5 models generated by the same server (Baker and Zhang), by our patch library. We see that in 28% and 23% of the cases, resp., the model that best fitted our patch library was indeed the model that had the best RMS score. The random expected value is 20%.

4. Conclusions and future work

This work presented a new library of surface patches analogous to the fragment libraries that had a considerable impact on computational structural biology over the last thirty years. As an initial test of the significance of this library we used it to compare patches taken from native structures to patches taken from decoys that were generated by state-of-the-art servers. Our results show that the clusters are indeed meaningful, and capture genuine aspects of native protein surfaces. Specifically, patches of decoys generated by servers are significantly different from patches of native proteins. Moreover, this difference has a predictive power allowing us to identify native protein structures within a set of server models. The ability to distinguish between different servers was an unpredictable but interesting result.

It should be noted that this phenomenon had little to do with the qualities of the models as measured by the standard RMS and GDT_TS scores. On the one hand, this means that we cannot reliably rank decoys. On the other hand, one may speculate that our library will shed light on inherent limitations of the current modeling techniques. Such limitations in the representation of surfaces may be overlooked by the current model assessment procedures. However they may drastically reduce the applicability of models for real life problems that often involve surface interactions. In this context it is very interesting that the Baker server, which apparently invests the most in structure refinement, appears further than native on average compared to the other four servers. The characterization of these discrepancies between model surfaces and the surfaces of native structures is an obvious direction to continue this study. We hope that it would lead to some insight about the limitations of current modeling procedures and eventually to better model building techniques.

A wide range of parameters and methods were tested in this work, such as the number of clusters, initialization techniques, various thresholds, etc. (see Table 4). Although some parameters generated better results than others, the tendency as shown in the results was always similar. Incorporating several parameter values, such as creating a number of libraries with different radii, thus creating patches with different sizes and coverage area still needs to be examined.

While the above results are intriguing, we believe that they are merely the tip of an iceberg. Protein structures are extremely complex entities and no single perspective exposes all their properties. In the past,

any new representation (e.g., fragments) opened a plethora of new directions for studies. We do hope that the same would happen here. Among the foreseeable future applications of our library we envision:

a. Studies of higher orders of surface organization. We would like to know whether certain patches typically prefer other patches in their vicinity.
b. Studies regarding functional inference from the identification of patches or groups of patches. One may speculate that certain patches would be enriched in proteins with specific GO annotation [55] or in regions involved in the binding of other proteins or nucleic acids or small ligands. For an intriguing demonstration of this direction see Figure 8.
c. Studies regarding the evolutionary conservation of these patches. Patterns of surface residues conservation proved very informative [56][57]. One may expect that the study of surface patches conservation will refine these results.
d. One of the most interesting questions to be asked relates to the patch composition of the different clusters. Is the composition itself meaningful? For example, do small clusters capture a unique surface motif of proteins? Can one find patches from active sites of different proteins in the same cluster? Can the clusters unravel functional connections between patches, and as a result, between proteins?
e. We believe that the current study is only the first step towards a better definition of surface patches. Due to time restrictions, several arbitrary decisions were made regarding the patch definition, the distance measure and its approximation, and the clustering technique and its parameters. Now that the feasibility of our approach has proven it seems appropriate to gradually review many of these decisions. We would like to suggest as a plausible target function for these refinement efforts, the ability of the patch library to reproduce the surfaces of proteins. A similar approach proved valuable in the study of protein fragments [1].

As presented above, future research still needs to be conducted in order to fully utilize the library. We expect that a high-quality and generic surface patch library will add a new set of building blocks to protein structure description, improve our protein prediction abilities, particularly on the surface and enhance our capability to determine the quality of predicted models.

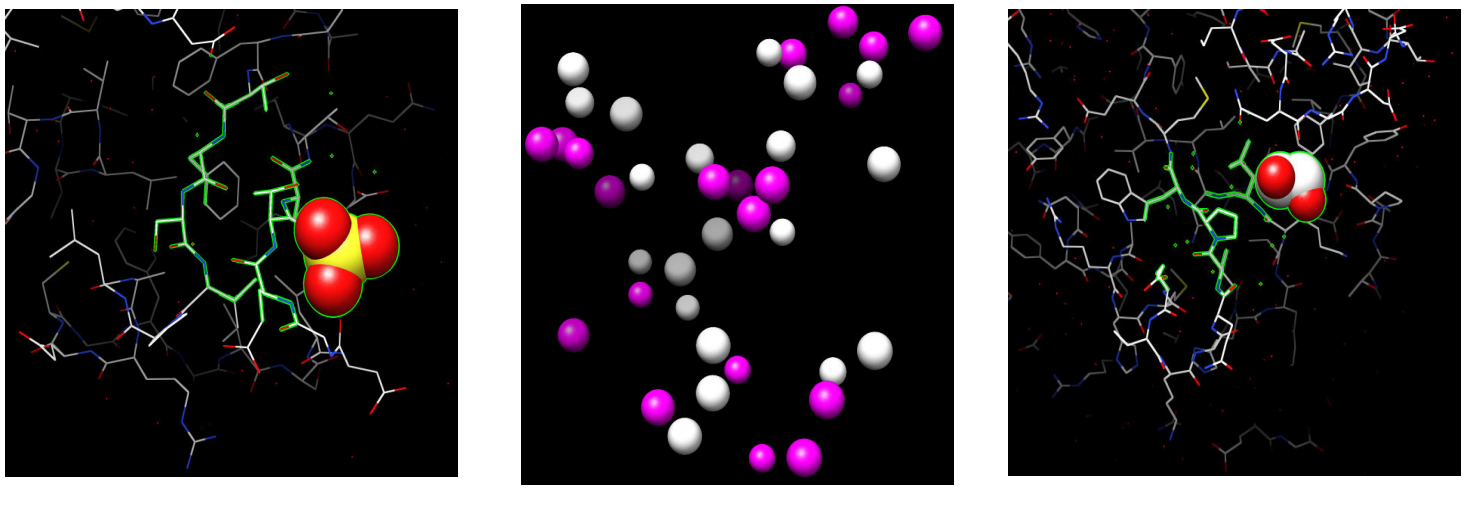

Figure 8: An interesting example that may demonstrate the relation between cluster composition and GO annotations was observed in a cluster containing 56 patches. Two members of the cluster, originated from DEACETOXYCEPHALOSPORIN C SYNTHASE (PDB code 1DCS), residue 258, seen on the left, and NITROGENASE MO-FE PROTEIN (PDB code 1QGU) residue 291, seen on the right, containing 21 and 22 atoms respectively, were found in a distance of 1.82 Å from each other. These two proteins are both classified as Oxidoreductases. Further, both of the patches were located in a binding site.

The left figure shows the patch, colored green, ligand sulfate ion (SO4) in sphere mode colored red and yellow. The right figure shows the patch, colored green, ligand 1,2-ETHANEDIOL in sphere mode colored red and white. The middle figure shows the two patches superimposed, using our distance function. The left patch is rendered white and the right patch is rendered purple.

References:

[1] R. Kolodny, P. Koehl, L. Guibas and Michael Levitt, "Small Libraries of Protein Fragments Model Native Protein Structures Accurately", *J. Mol. Biol*. (2002) 323, 297-307.

[2] L. Baldacci, M. Golfarelli,A. Lumini, S. Rizzi, "Clustering techniques for protein surfaces", *Pattern Recognition* 39 (2006) 2370-2382.

[3] L. Baldacci, M. Golfarelli, A. Lumini, S. Rizzi, "A Template-Matching Approach for Protein Surface Clustering", IEEE (2006), *ICPR06* III, 340-343.

[4] L. Baldacci, M. Golfarelli, "Mining Complex Patterns from Protein Surfaces", IEEE (2005), *DEXA Workshops* 590-594.

[5] S. Jones and J. M. Thornton, "Analysis of Protein-Protein Interaction Sites using Surface Patches", *J. Mol. Biol.* (1997) 272, 121-132.

[6] S. Jones, J.M Thornton, "Prediction of protein-protein interaction sites using patch analysis", *J. Mol. Biol.* (1997) 272, 133-143.

[7] Y. Murakami and S. Jones, "SHARP: protein–protein interaction predictions using patch analysis", *Bioinformatics* (2006) Vol. 22 no. 14, 1794-1795.

[8] TA. Jones and S. Thirup. "Using known substructures in protein model building and crystallography", EMBO J. (1986) 5, 819-822.

[9] R. Unger, D. Harel, S. Wherland, & J. L. Sussman, (1989). "A 3D building blocks approach to analyzing and predicting structure of proteins". *Proteins: Struct. Funct. Genet.* 5, 355-373.

[10] J.M. Chandonia, G. Hon, N.S. Walker, L. Lo Conte, P. Koehl, M. Levitt, SE Brenner. "The ASTRAL compendium" (2004), *Nucleic Acids Research 32:*D189-D192.

[11] B. Lee and F.M.Richards, "The interpretation of protein structure: estimation of static accessibility". *J. Mol. Biol.*, (1971), **55**, 379-400.

[12] PROGEOM http://nook.cs.ucdavis.edu/~koehl/ProShape/

[13] S. Lloyd, Last square quantization in PCM's. Bell Telephone Laboratories Paper (1957). Published in journal much later: S. P. Lloyd. "Least squares quantization in PCM. Special issue on quantization", IEEE Trans. Inform. Theory (1982), 28:129-137.

[14] A. Via, F. Ferre, B. Brannetti, and M. Helmer-Citterich, “Protein surface similarities: a survey of methods to describe and compare protein surfaces,” *Cell. Mol. Life Sci.* (2000), vol. 57, pp. 1970-1977.

[15] S. J. Pickering, A. J. Pulpitt, N. Efford, N. D. Gold, and D. R.Westhead, “AI-based algorithms for protein surface comparisons,” *Comput. Chem.*, (2001), vol. 26, pp. 79-84.

[16] M. Drabikowski, S. Nowakowski, J. Tiuryn, “Library of local descriptors models the core of proteins accurately” *Proteins,* (2007), **69**: 499-510.

[17] D. Duhovny, , R. Nussinov, & H. J. Wolfson, “Efficient unbound docking of rigid molecules”. In Workshop on Algorithms in Bioinformatics (Guigo, R. & Gusfield, D., eds), (2002), vol. 2452, pp. 185-200, LNCS, Springer.

[18] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig et al. “The Protein Data Bank”. *Nucl. Acids Res.,* (2000), 28, 235-242.

[19] A. G. Murzin, S. E. Brenner, T. Hubbard & C. Chothia. “SCOP: a structural classification of proteins database for the investigation of sequences and structures”. *J. Mol. Biol.*, (1995), 247, 536-540.

[20] S. E. Brenner, P. Koehl and M. Levitt. “The ASTRAL compendium for protein structure and sequence analysis” *Nucl. Acids Res*., (2000), 28, 254-256.

[21] Kabsch, Wolfgang, "A solution of the best rotation to relate two sets of vectors", *Acta Crystallographica,* (1976), 32:922-923.

[22] J.B. MacQueen "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, (1967), 1:281-297.

[23] D. Hochbaum, D. Shmoys. “A best possible heuristic for the k-center problem”, *Mathematics of Operations Research*, (1985), 10(2):180-184.

[24] D. Arthur and S. Vassilvitskii. “kmeans++: The advantages of careful seeding”. *In ACM-SIAM Symposium on Discrete Algorithms*, (2007), pages 1027-1035.

[25] B. Lee and FM. Richards. “The interpretation of protein structure: estimation of static accessibility”. *J. Mol. Biol.*, (1971), 55, 379-400.

[26] J. Greer, B. Bush, “Macromolecular Shape and SurfaceMaps by Solvent Exclusion”, *Proceedings of the National Academy of Sciences USA,* (1978), 75, 303-307.

[27] M. L. Connolly, “Analytical Molecular Surface Calculation”, *Journal of Applied Crystallography,* (1983), 16, 548-558.

[28] F. Jiang, S. H. Kim ‘‘Soft docking: matching of molecular surface cubes”. *J. Mol. Biol.*, (1991), 219: 79-102.

[29] B. Shoichet, I. D. Kuntz “Protein docking and complementarity”. *J. Mol. Biol.,* (1991), 221: 327-346.

[30] M. H. Citterich, A. Tramontano “PUZZLE: a new method for automated protein docking based on surface shape complementarity”. *J. Mol. Biol*., (1994), 235: 1021-103.

[31] G. Ausiello, G. Cesareni and Helmer M. Citterich “ESCHER: a new docking procedure applied to the reconstruction of protein tertiary structure”. *Proteins*, (1997), 28: 556-567.

[32] A. Shulman-Peleg, M. Shatsky, R. Nussinov, H.J. Wolfson “Prediction of interacting single-stranded RNA bases by protein-binding patterns”. *J Mol Biol.*, (2008), 379:299-316.

[33] J. Liang, H. Edelsbrunner, P. Fu, P.V. Sudhakar, and S. Subramaniam, “Analytical shape computation of macromolecules” I and II. *Proteins: Struct. Funct. Genet.*, (1998), 33, 1-17 and 18-29.

[34] B. Von Freyberg, T.J. Richmond, and W. Braun, “Surface area included in energy refinements of proteins: a comparative study on atomic solvation parameters”. *J. Mol. Biol.*, (1993), 233, 275-292.

[35] R. Fraczkiewicz, and W. Braun, “Exact and efficient analytical calculation of the accessible surface area and their gradient for macromolecules”. *J. Comput. Chem.*, (1998), 19, 319-333.

[35] D. Chivian, D.E. Kim, L. Malmstrom, J. Schonbrun, CA. Rohl, and D. Baker. ”Prediction of CASP6 structures using automated Robetta protocols.” *Proteins 61 Suppl,* (2005), 7:157-66.

[36] D. Fischer, H. Wolfson, S.L. Lin, and R. Nussinov. “Three-dimensional, sequence order-independent comparison of a serine protease against the crystallographic database reveals active site similarities: Potential implications to evolution and to protein folding.” *Protein Sci,* (1994), 3: 769-778.

[37] T.A. Binkowski, A. Joachimiak. “Protein functional surfaces: global shape matching and local spatial alignments of ligand binding sites.” *BMC Struct. Biol* (2008) 8:45.

[38] K.F. Han and D. Baker, “Global properties of the mapping between local amino acid sequence and local structure in proteins.” *Proc. Natl Acad. Sci.,* (1996), *USA*, 93, 5814-5818.

[39] Y. Ye, L. Jaroszewski, W. Li and A. Godzik “A segment alignment approach to protein comparison.” *Bioinformatics*, (2003), 19, 742-749.

[40] I. Friedberg, A. Godzik “Connecting the protein structure universe by using sparse recurring fragments.” *Structure (Camb),* (2005), **13:**1213-24.

[41] M. Levitt “Accurate Modeling of Protein Conformation by Automatic Segment Matching”. J. *Mol. Biol.,* (1992), 226: 507-533.

[42] Q. Le, G. Pollastri1, P. Koehl. “Structural Alphabets for Protein Structure Classification: A Comparison Study.” *J. Mol. Biol*. (2009) 387, 431-450.

[43] P. Zhou, J. Zou, F. Tian, Z. Shang “Geometric Similarity Between Protein–RNA Interfaces.” Journal of Computational Chemistry. Published Online**:** 27 Apr 2009.

[44] LS. Lin, R. Nussinov, D. Fischer, H.J. Wolfson. “Molecular surface representation by sparse critical points.” Proteins, (1994); 18:94-101.

[45] R. Norel, D. Fischer, H. Wolfson, R. Nussinov “Molecular surface recognition by a computer vision based technique.” Prot Eng, (1994); 7:39-46.

[46] M.R. Chernick. “Bootstrap Methods, A Practitioner’s Guide*”* Wiley, (1999), New York.

[47] F. J. Massey, "The Kolmogorov-Smirnov Test for Goodness of Fit." *Journal of the American Statistical Association*. Vol. 46, No. 253, (1951), 68-78.

[48] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, “Basic local alignment search tool” *J. Mol. Biol.,* (1990), **215**, 403.

[49] Y. Zhang “I-TASSER server for protein 3D structure prediction**.”** *BMC Bioinformatics* (2008), **9:**40.

[50] R. L. Thorndike. "Who Belong in the Family?" *Psychometrika* (1953) **18** (4).

[51] C. Goutte, L. Hansen, M.G. Liptrot, E. Rostrup. "Feature-Space Clustering for fMRI Meta-Analysis". *Human Brain Mapping* **13** (2001) (3): 165–183.

[52] C. Sugar, G. James. "Finding the number of clusters in a data set: An information theoretic approach". *Journal of the American Statistical Association* (2003) **98**: 750–763.

[53] P. Rousseuw. "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". *Computational and Applied Mathematics* (1987) **20**: 53–65.

[54] R. Lleti, M.C. Ortiz, L.A. Sarabia, M.S. Sánchez. "Selecting Variables for k-Means Cluster Analysis by Using a Genetic Algorithm that Optimises the Silhouettes". *Analytica Chimica Acta* (2004) **515**: 87–100.

[55] M. Ashburner,. et al. "Gene ontology: tool for the unification of biology" The Gene Ontology Consortium. Nature Genet. (2000) 25, 25–29.

[56] F. Glaser, T. Pupko, I. Paz, R.E. Bell, D. Bechor, E. Martz. And N. Ben-Tal. "ConSurf: Identification of Functional Regions in Proteins by Surface-Mapping of Phylogenetic Information." (2003) *Bioinformatics* 19:163-164.

[57] Detection of functionally important regions in "hypothetical proteins" of known structure. G. Nimrod, M. Schushan, DM Steinberg, N. Ben-Tal. Structure. (2008).16(12):1755-63.

[58] T. Nakatsu, H. Kato, J. Oda. "Crystal structure of asparagine synthetase reveals a close evolutionary relationship to class II aminoacyl-tRNA synthetase" Nat Struct Biol. (1998); 5(1):15-9.

Supplementary

System Parameters (Table 4)

Parameter name	Function	Range tested	Selected value
R	Radii of a patch	7Å	7Å
r	Radii of patch inner sphere	4Å	4Å
α	The percent from which we define a surface atom as being on the surface	2%,5%,10% and 30%	10%
outlier_threshold	A patch is considered an outlier if it is distant (more than 1.8Å) from more than outlier_threshold % patches.	90%, 95% and 97.5%	90%
K	Numbers of clusters patches were portioned to.	100, 200, 250, 300, 350	350

Training set (PDB name, structure resolution, SPACI score) (Table 5):

d1gci__	0.78	1.33	d3lzt__	0.92	1.15	d1b0ya_	0.93	1.07	D1aho__	0.96	1.04	d3sil__	1.05	0.99
d1cbn__	0.83	1.23	d1bxoa_	0.95	1.10	d1byi__	0.97	1.07	D1cxqa_	1.02	1.03	d2igd__	1.10	0.98
d3pyp__	0.85	1.20	d2fdn__	0.94	1.10	d1cex__	1.00	1.07	D2erl__	1.00	1.02	d1qj4a_	1.10	0.94
d1rb9__	0.92	1.17	d7a3ha_	0.95	1.09	d1ixh__	0.98	1.06	D1mfma_	1.02	1.01	d5pti__	1.00	0.92
d2pvba_	0.91	1.15	d1nls__	0.94	1.07	d1a6m__	1.00	1.05	D1lkka_	1.00	1.00	d1rgea_	1.15	0.92
d1bkra_	1.10	0.92	e1pid1b	1.30	0.70	d1rie__	1.50	0.63	D1kapp1	1.64	0.59	d1qhfa_	1.70	0.56
d1nkd__	1.07	0.92	e1pid1a	1.30	0.70	d3ezma_	1.50	0.63	d2cpl__	1.63	0.59	d1dhn__	1.65	0.56
d1swua_	1.14	0.91	d3euga_	1.43	0.69	d1bfd_2	1.60	0.63	d1b6a_2	1.60	0.59	d2ahja_	1.70	0.56
d1a7s__	1.12	0.88	d3vub__	1.40	0.68	d1ra9__	1.55	0.62	d1b6a_1	1.60	0.59	d3stda_	1.65	0.56
d1mun__	1.20	0.88	d1qfma1	1.40	0.67	d1dfma_	1.50	0.62	d3grs_3	1.54	0.59	d1yvei1	1.65	0.56
d1vfya_	1.15	0.85	d1bgf__	1.45	0.67	d1a4ia2	1.50	0.62	d1qqqa_	1.50	0.59	d1pda_2	1.76	0.56
d1jhga_	1.30	0.83	d1aba__	1.45	0.67	d1c1ka_	1.45	0.62	d1mrj__	1.60	0.59	d1vcc__	1.60	0.56
d1d4oa_	1.21	0.82	d1qtsa2	1.40	0.67	d1byqa_	1.50	0.62	d1aop_3	1.60	0.59	d1pdo__	1.70	0.56
d1qu9a_	1.20	0.82	d1sgpi_	1.40	0.67	d1aie__	1.50	0.62	d1php__	1.65	0.59	d1utea_	1.55	0.55
d3chbd_	1.25	0.82	d1di6a_	1.45	0.67	d1dpta_	1.54	0.62	d1csh__	1.60	0.58	d1ush_1	1.73	0.55
d1mroa1	1.16	0.81	d1utg__	1.34	0.67	d1bx4a_	1.50	0.62	d1t1da_	1.51	0.58	d1cjca2	1.70	0.55
d1ifc__	1.19	0.81	d7atja_	1.47	0.66	d1qgua_	1.60	0.62	d1ajsa_	1.60	0.58	d1b2pa_	1.70	0.55
d7rsa__	1.26	0.80	d1yge_2	1.40	0.66	d1c3wa_	1.55	0.62	d1qs1a1	1.50	0.58	d3btoa1	1.66	0.55
d1dcs__	1.30	0.79	d1yge_1	1.40	0.66	d1hfes_	1.60	0.61	d1a1ia1	1.60	0.58	d1kpta_	1.75	0.55
d2pth__	1.20	0.79	d1tc1a_	1.41	0.66	d1hfel1	1.60	0.61	d1smd_1	1.60	0.58	d1gsoa3	1.60	0.55
d1amm_1	1.20	0.78	d1mla_1	1.50	0.66	d1qgwa_	1.63	0.61	d1b8za_	1.60	0.58	d1gsoa2	1.60	0.55
d2lisa_	1.35	0.78	d1poa__	1.50	0.66	d1orc__	1.54	0.61	d1ay7b_	1.70	0.58	d1gsoa1	1.60	0.55
d1cy5a_	1.30	0.77	d2cba__	1.54	0.65	d1qsaa1	1.65	0.61	d1fmk_3	1.50	0.58	d1dmr_2	1.82	0.55
d1aac__	1.31	0.77	d3pte__	1.60	0.65	d1dpsa_	1.60	0.61	d1phc__	1.60	0.58	d1atza_	1.80	0.54
d1qdda_	1.30	0.76	d1pina_	1.35	0.65	d1nox__	1.59	0.61	d1qgxa_	1.60	0.57	d1fnd_2	1.70	0.54
d1dg6a_	1.30	0.76	d1g3p_1	1.46	0.65	d1b3aa_	1.60	0.61	d1d3va_	1.70	0.57	d1fnd_1	1.70	0.54
d1msi__	1.25	0.75	d1cyo__	1.50	0.64	d1cipa1	1.50	0.61	d7odca1	1.60	0.57	d1a44__	1.84	0.54
d1qksa2	1.28	0.75	d1qrea_	1.46	0.64	d1kpf__	1.50	0.60	d1vns__	1.66	0.57	d1vhh__	1.70	0.54
d256ba_	1.40	0.73	d1dgfa_	1.50	0.64	d1lam_1	1.60	0.60	d1ctf__	1.70	0.57	d1aoha_	1.70	0.54
d1bi5a1	1.56	0.72	d1whi__	1.50	0.64	d1krn__	1.67	0.60	d1czfa_	1.68	0.57	d1doza_	1.80	0.54
d1rhs__	1.36	0.72	d1ezm_2	1.50	0.64	d1gai__	1.70	0.60	d1rzl__	1.60	0.57	d1db1a_	1.80	0.54
d1qh4a2	1.41	0.72	d1ezm_1	1.50	0.64	d1bfg__	1.60	0.60	d1tx4a_	1.65	0.57	d1thw__	1.75	0.54
d1qh4a1	1.41	0.72	d1dcia_	1.50	0.64	d1d7pm_	1.50	0.60	d1ako__	1.70	0.57	d1tfe__	1.70	0.54
d1qaua_	1.25	0.72	d1b67a_	1.48	0.64	d1moq__	1.57	0.60	d3cla__	1.75	0.57	d1svy__	1.75	0.54
d3ebx__	1.40	0.71	d1qh5a_	1.45	0.63	d1qq5a_	1.52	0.60	d1burs_	1.80	0.56	d1mjha_	1.70	0.54
d2eng__	1.50	0.71	d1b4va2	1.50	0.63	d1ubpc1	1.65	0.60	d1kid__	1.70	0.56	d1bm8__	1.71	0.54
d1qhva_	1.51	0.70	d1b4va1	1.50	0.63	d1ubpa_	1.65	0.60	d2cpga_	1.60	0.56	d2gsta1	1.80	0.53
d2end__	1.45	0.70	d8abp__	1.49	0.63	d3cyr__	1.60	0.59	d2bbkl_	1.75	0.56	d1pcfa_	1.74	0.53
d1bsma2	1.35	0.70	d1ah7__	1.50	0.63	d2ilk__	1.60	0.59	d1qipa_	1.72	0.56	d1mtyg_	1.70	0.53
d1bsma1	1.35	0.70	d1ptf__	1.60	0.63	d1ppn__	1.60	0.59	d1ttba_	1.70	0.56	d1iiba_	1.80	0.53

Test set (Table 6)

Target name	Length	Organism	PDB code	Target name	Length	Organism	PDB code
T0388	174	Homo sapiens	3cyn	T0440	275	Listeria monocytogenes	3dcp
T0389	153	Homo sapiens	2vsw	T0444	326	Homo sapiens	2vux
T0390	182	Homo sapiens	3czu	T0447	542	Thermotoga maritima	3do6
T0391	157	Mouse	3d89	T0448	232	Lactobacillus plantarum	3dc7
T0392	109	Homo sapiens	2vsv	T0449	307	Lactobacillus acidophilus	3dcd
T0394	275	homo sapiens	3dcy	T0450	561	Bacillus halodurans	3da1
T0395	304	Candida glabrata		T0451	133	Anabaena variabilis	3dmc
T0396	105	African Swine Fever Virus		T0453	91	Chromobacterium violaceum	3ded
T0399	206	Streptococcus mutans	3d4e	T0455	145	Enterococcus faecalis V583	3ddv
T0400	162	Staphylococcus aureus		T0458	107	Streptomyces avermitilis	3dex
T0401	143	Bacteroides fragilis	3d5p	T0459	111	Thermoplasma volcanium	3df8
T0402	139	LISTERIA INNOCUA	3db0	T0461	189	Homo sapiens	3dh1
T0404	110	Anabaena variabilis	3dfe	T0463	219	Bacteroides thetaiotaomicron	3dhn
T0406	167	BACILLUS CEREUS	3di5	T0465	157	Bacillus subtilis	3dfd
T0408	104	Methanococcus jannaschii	3d7i	T0477	242	Homo sapiens	3dkp
T0409	103	Nitrosomonas europaea	3d0f	T0479	134	Bordetella parapertussis	3dkz
T0411	139	Saccharomyces cerevisiae	3d1p	T0481	154	Bacillus subtilis	3dka
T0412	178	Acinetobacter sp. ADP1	3d3o	T0483	335	Homo sapiens	3dls
T0413	304	Bordetella parapertussis	3d0k	T0485	218	Methanococcus maripaludis	3dlc
T0414	140	Clostridium acetobutylicum	3d0j	T0486	287	Homo sapiens	2vx2
T0415	109	Bacillus cereus	3d6w	T0488	95	Homo sapiens	2vwr
T0417	189	Bordetella parapertussis	3d3s	T0489	266	Klebsiella pneumoniae	3dl1
T0420	189	Sulfolobus solfataricus	3d5n	T0490	369	Bordetella pertussis	3dme
T0421	300	Sinorhizobium meliloti	3czq	T0491	115	Bordetella pertussis	3dm4
T0423	156	Agrobacterium tumefaciens str. C58	3d01	T0493	174	Lactobacillus plantarun	3dmn
T0425	181	Neisseria meningitidis MC58	3czx	T0494	382	Homo sapiens	2vx3
T0426	283	Homo sapiens	3da2	T0495	150	Silicibacter pomeroyi	3dnx
T0428	267	Cryptosporidium parvum	3d8h	T0497	146	XANTHOMONAS CAMPESTRIS	3dmb
T0432	130	Homo sapiens	3dai	T0502	105	Methanocaldococcus jannaschii	3dm3

T0433	199	Listeria innocua	3d7l	T0503	191	Cytophaga hutchinsonii	3dn7
T0434	205	Homo sapiens	3dal	T0507	148	Archaeoglobus fulgidus	3do8
T0435	151	Homo sapiens	3db5	T0508	197	Thermoplasma volcanium	3dou
T0436	419	Corynebacterium diphtheriae	3d6k	T0509	213	Corynebacterium glutamicum	3dr5
T0512	328	Bacteroides uniformis	3dsm	T0511	271	Xanthomonas campestris	3e03
				T0514	145	BARTONELLA HENSELAE	3dtd