Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 39 (2006) 2370–2382
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Clustering techniques for protein surfaces

L. Baldacci, M. Golfarelli*, A. Lumini, S. Rizzi

DEIS, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

Received 5 July 2005; received in revised form 14 February 2006; accepted 22 February 2006

Abstract

Though most approaches to protein comparison are based on their structure, several studies produced evidence of a strict correlation between the surface characteristics of proteins and the way they interact. Surface-based techniques for protein comparison typically require applying clustering algorithms to the punctual 3D description of the surface in order to produce a compact surface representation, capable of effectively condensing its description. In this paper, we propose a formalization of the requirements for surface clustering in the biochemical context and present two different clustering techniques that meet them, based, respectively, on region-growing and on an original template matching algorithm. We discuss the validity of these techniques with the support of tests performed on a set of about one hundred protein models generated by punctual mutations of four structurally characterized proteins. Finally, an analysis is made of how different factors impact on the effectiveness of clustering in capturing surface similarities.

Keywords: Clustering; Region growing; Template matching; Protein surface

1. Introduction

Understanding which characteristics of proteins have most impact on their functional role is one of the main challenges of the post-genomic era. In this direction, the techniques for structural comparison between proteins devised so far have given a very relevant contribution [1–3]. On the other hand, protein function occurs predominantly on or near the protein surface [4], so comparison of protein surfaces may reveal functional relationships not found with structural comparison. In fact, two proteins with different structural fold may present similar chemical properties over their surface. For example, trypsin-like and subtilisin-like serine protease families have different folds but present strong similarities between their active sites [5].

Remarkable applications for surface-based comparison are *registration*, that is the problem of finding whether two proteins have similar shapes [4,6,7], *biomolecular docking*, that studies under which conditions two proteins can interact with each other and create a stable energetic contact [8], *similarity search* on databases of surface patches with known functionalities [9,10], and *classification*, aimed at grouping similar proteins [6,11,12].

Several surface-based techniques for protein comparison depend on a compact surface representation capable of effectively condensing the description of its properties [6,7,12]. The choice of the level of detail to be used when representing the surface is guided by the following considerations:

- The stability and the strength of inter-protein interaction depend on the extension of the interaction areas. An analysis made on the SURFACE database [10], that includes a wide set of active sites, reveals that the smallest patches for significant interactions include two or three amino acids.
- Proteins could show a high flexibility that allows the shape of their surface to be partially modified and adapted for stronger interactions. Thus, giving too much relevance to spatial details may be misleading.

* Corresponding author. CdL in Scienze dell'Informazione, Via Sacchi 3, 47100 Cesena, Italy. Tel.: +39 0547 338862; fax: +39 0547 338890.
E-mail address: golfare@csr.unibo.it (M. Golfarelli).

0031-3203/$30.00
doi:10.1016/j.patcog.2006.02.024

- Surface points belonging to the same patch should have homogeneous features, from both the chemical and geometrical points of view.

A suitable surface representation is typically obtained by applying clustering techniques to the punctual 3D description of the surface in order to generate a set of homogeneous patches. Unfortunately, the previous works on patch-based protein comparison [4,6,7] mainly focus on the matching methods and, from our point of view, suffer from three limitations: (1) there is no discussion of which requirements should drive clustering in order to maximize its effectiveness; (2) only geometrical properties of the protein surfaces are considered (e.g., concavity and convexity); and (3) no details about the clustering algorithms are given.

In this paper, we address the clustering problem from all these points of view, by proposing the following main contributions:

- A discussion of the requirements for protein surface clustering is made, and a set of target functions that capture them is introduced. Though the application domain requires that patches have homogeneous features, have a regular shape, are "not too small", and achieve a large coverage of the protein surface, no precise indication is available about the relative importance of these factors. Therefore, it is not possible to define a single target function to be directly optimized.
- We propose a multi-feature approach that, besides geometrical properties of patches, also considers chemical ones, namely electrostatic potential and hydrophobicity. Note that, though features range within continuous domains of values, a patch is significant from the biochemical point of view even if it is categorized according to a non-linear discretization of the domain made for each feature (e.g., as far as the electrostatic potential is concerned, a patch may be categorized as positive, neutral, or negative).
- Two different clustering techniques are proposed and compared. The first one gives maximum priority to building homogeneous patches; it is based on a region-growing algorithm, properly adapted to the peculiarities above. Conversely, the second one gives maximum priority to building regularly shaped patches and is based on an original template matching algorithm. Both clustering algorithms are parametric w.r.t. the selection of the chemical surface properties.
- Through an ad hoc dataset the optimality, robustness and stability of the algorithms are evaluated and an analysis is made of how different factors impact on the effectiveness of clustering in capturing surface similarities. Tests have been carried out within the surface-based classification framework we are developing [12].

The rest of the paper is structured as follows. Section 2 discusses some relevant literature related to protein surface comparison and clustering of 3D surfaces. Section 3 shows how clustering is framed within the approach to protein classification we are pursuing, while Section 4 describes in detail the two approaches to clustering we propose. Section 5 reports and discusses the results of the tests we carried out, and Section 6 contains the final evaluation of the approaches and the conclusions.

2. Related literature

2.1. Protein surface comparison

Protein surface comparison has been gaining more and more significance in the last decade. This task, that consists in highlighting similar surface portions on different molecules, may sometimes point out different structures or sequences coming together in a unique active site having a common function [5,13].

The choice of the surface description is doubtlessly critical, since a satisfactory trade-off should be reached between the representation grain and the performance of the comparison algorithm. Mainly three approaches have been devised to describe the molecular surface, namely *mesh-based*, *atom-based*, and *patch-based*, discussed in the following.

As for mesh-based methods, several of them use the so-called Connolly algorithm [14]. Here, the surface is traced by a water-sized probe sphere rolled over the atoms of the molecule; the solvent accessible surface is obtained by the set of points touched by the center of the probe. Two effective comparison methods based on the Connolly surface are proposed in Refs. [7,9]; in both cases, similarity is determined by comparing the spatial arrangement of the normal vectors and the values of the local properties at the surface nodes.

Atom-based methods define the molecular surface as a significant subset of the solvent exposed atoms. For example, in Ref. [11] the author defines the α-surface as the set of atoms touched by a probe of given radius. This description is the starting point to discover surface patterns within a database of proteins aimed at classifying them.

In patch-based methods, the grain of the surface is increased by considering the set of exposed amino acids or a compact representation obtained by segmenting the surface into homogeneous regions. In Ref. [10], the authors describe a method for the function-related annotation of protein structure by means of the detection of local structural similarity through a library of annotated functional sites. A graph-based method for protein comparison and classification using a patch representation of the surface has been described in Ref. [6]. Here, comparison relies on matching surface graphs, whose nodes are concave, convex, and toroidal patches extracted from the Connolly surface. The comparison method proposed in Ref. [15] finds maximal common subgraphs on surface graphs whose nodes correspond to the centers of slightly overlapping circular patches.

In most works on docking, emphasis is given to effectively represent the protein surface and to study efficient methods for searching patterns compatible with the probe. Only a few of them deal with the problem of segmenting protein surface into homogenous patches. In Ref. [16], a region growing method is used to progressively group similar adjacent surface elements. In Ref. [17] surface points are quantized based on their geometrical properties and their hydrophobicity, then a complex region growing technique is applied to obtain compact regions with maximal extension. In Ref. [18] a Morse–Smale decomposition based on Formans discrete Morse theory is used to partition the protein surface into regions of homogeneous curvature. Though these approaches could, in principle, be used also in the context of protein classification, this context actually presents more specific requirements; besides, none of these papers includes a general analysis of the clustering issues related to docking applications.

2.2. Surface clustering

In Section 2.1 a number of comparison methods based on a compact representation of the surface have been introduced. All these works focus on comparison techniques but do not study the problem of surface clustering in depth. On the other hand, there has been considerable research work on clustering algorithms for general 3D surfaces; most methods are known as *mesh partitioning approaches* and concern computer graphics applications. One example is surface simplification [19], aimed at radically reducing the amount of data used for surface representation without a considerable loss of details. Several other applications of mesh partitioning are related to decomposition of 3D CAD models aimed at speeding up searches on model databases [20].

Most approaches proposed in the different domains are boundary-based, i.e., they tend to partition the surface along boundaries with high (positive or negative) curvature. For instance, in Ref. [21] the morphological watershed approach proposed for image segmentation is generalized to 3D surfaces: a surface is segmented where sharp differences in the surface normal create a boundary, without requiring boundaries to be located a priori, e.g. by means of second-order derivatives of curvature. The watershed strategy relies on a top–down approach which starting from a point moves along the deepest descent, joining points together until it reaches a minimum. Then, a post-processing step resolves over-segmentation. Wu and Levine [22] propose another boundary-based method, starting from the simulated distribution of electrostatic charge across the surface of a mesh. Their approach is based on a human vision theory stating that human perception defines part boundaries along lines of maximum negative curvature. Overall, the main drawback of boundary-based approaches is the poor segmentation of areas with low contrast boundaries, due to the fact that the boundaries are naturally placed in regions that present maximum curvature. This makes such approaches unsuitable for protein surface segmentation, where the boundaries between regions are not necessarily required to be points of maximum curvature.

Another relevant class of segmentation methods is based on region growing [23]. In Ref. [24] nodes are first classified using their discrete curvature values, then connected triangle regions are extracted via a region growing process, and finally similar regions are merged by a region adjacency graph to obtain final patches. In Ref. [25], segmentation is carried out in three phases: segment initialization based on region growing, computation of segment centers, assignment of nodes to segments, and optional segment merging. Another simple algorithm for mesh decomposition based on curvature analysis joins detection of boundary, meant as points with highly negative curvature, and region growing [26]. Region growing algorithms are very efficient and have also been applied in bioinformatics to solve the docking problem [25]; nevertheless, they cannot immediately be used in our context since they operate on a single surface feature.

3. A framework for clustering

The present work is framed within a larger research project aimed at defining a classification of proteins based on surface properties. In this section, we summarize our approach to classification, that will be used in Section 5 to evaluate the results of clustering. In fact, the correctness of the classification obtained on a set of proteins indirectly measures how effectively clustering represents the protein surface.

As sketched in Fig. 1, our approach consists of four steps: (1) determine, on each protein of a given dataset, a collection of homogeneous and connected surface regions (*patches*) by means of a clustering algorithm; (2) synthetically represent each protein by a spatial graph of patches; (3) find frequent patterns of patches through mining techniques; and (4) classify proteins based on the frequent patterns their surface presents. Using patterns of patches instead of single patches enables a more adherent modeling of the protein surface. Besides, determination of patches does not depend on already known functional meanings, thus clearing the way towards new classifications not constrained by previous knowledge.

The approach takes in input a set of proteins described in the PDB format [28], then the solvent-accessible protein surfaces and their electrostatic potentials are calculated using the MOLMOL program [27]. Finally, curvature and hydrophobicity are calculated for each mesh (see Section 4.1). The first step starts from this punctual representation of the surface and, by means of the clustering algorithms described in this work, delivers a more compact representation consisting of a set of homogeneous and connected patches.

The protein properties do not only depend on the set of patches characterizing their surface, but also on the relative positioning and orientation of these patches. Thus, each

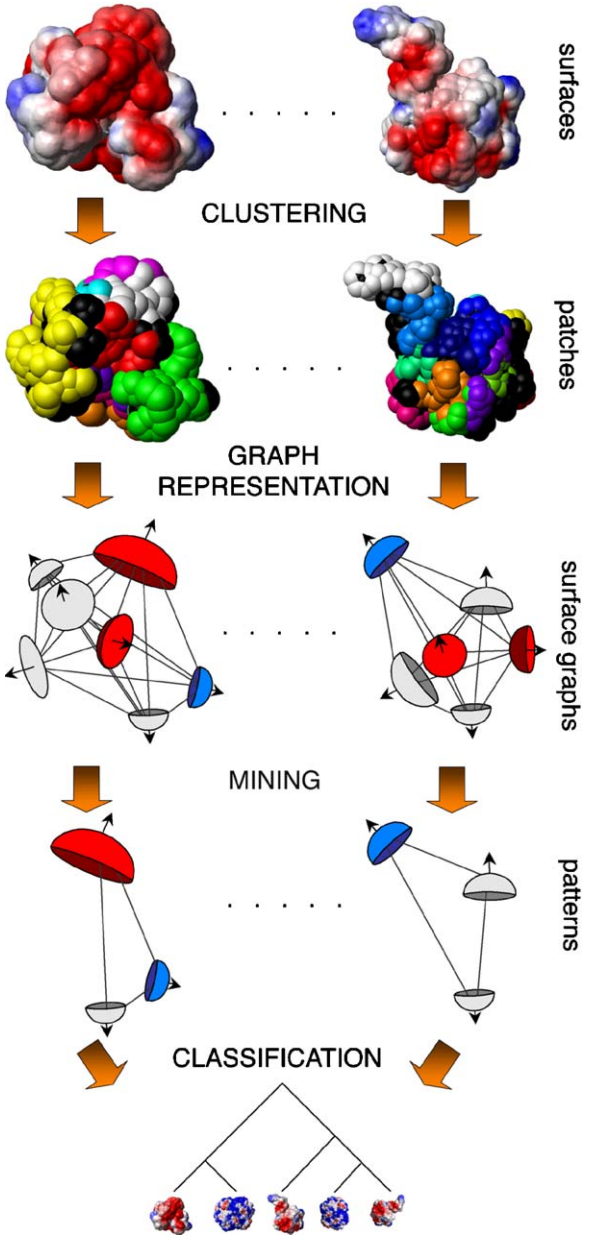

Fig. 1. Overview of surface-based protein classification (parts of this figure were prepared with the program MOLMOL [27]).

protein is compactly represented by a *surface graph* whose nodes are the patches obtained by clustering, each described by the average value of the features and by its area, and where the arc connecting two patches expresses their relative position in the 3D space as described in Ref. [12].

The final step before classification is the extraction of frequent surface patterns from the set of proteins. A *pattern* on a protein is a subgraph of its surface graph, thus it models a set of patches and their relative spatial placement. Extracting frequent patterns is challenging since two proteins never exhibit an *identical* pattern, so a similarity function involving both the local features of patches and their relative placement has to be defined. In Ref. [12] we proposed a level-wise mining algorithm that iteratively determines frequent patterns made up of an increasing number of patches.

Finally, classification groups together proteins that share common patterns into a dendrogram, which allows the domain experts to evaluate the ability of the approach to characterize protein similarities at both coarse and fine granularity levels. We adopt a hierarchical technique that, starting from clusters composed by a single protein, progressively merges the two most similar ones according to the complete-link approach [29]. Similarity is based on the number of shared patterns; the larger the pattern, the higher the contribution to the score function.

4. Protein surface clustering

As already mentioned in Section 1, the clustering of a protein surface must satisfy some contrasting requirements in order to be effective from a biochemical point of view:

1. A patch should be connected and cover at least 10 surface atoms. This value is an estimate of the minimum number of exposed atoms included in the smallest patches for significant interactions (see Section 1).
2. A patch should present homogeneous values for surface features with reference to their discretization. In other words, all the points belonging to a patch should ideally fall in the same category for each feature (e.g., positive electrostatic potential), though they are not required to yield the same value (e.g., the electrostatic potential within a positive patch may range from 0.4 to 0.9). Details on the representation of surface features and their discretization are given in Section 4.1.
3. A patch should have a regular shape. Though patches with irregular shape may lead to a better coverage of the protein surface, those with regular shape better characterize specific regions of the protein and promise to guarantee higher robustness.
4. The union of the patches should achieve a high percentage coverage of the protein surface.

While requirement (1) is considered to be necessary, there is no a priori clue about the relative importance of requirements (2)–(4), whose quantification is given in Section 4.2. Thus we experiment two different clustering techniques, both constrained to meet requirement (1). The first technique, described in Section 4.3, is based on region growing and builds homogeneous patches with good coverage but possibly irregular shapes; the second, described in Section 4.4, uses a template matching algorithm to build circular patches whose feature homogeneity is higher than a threshold.

4.1. Surface representation

A fine-grained 3D representation of a protein surface is usually given in terms of triangular meshes. With reference

to our approach to clustering, the mesh granularity is too fine since small-scale local variations of feature values may be misleading. Thus, also considering that—according to domain experts—the meshes related to the same atom should never be split into separate patches, and that requirement (1) asks for building patches that include at least 10 atoms, we decided to use a coarser representation based on surface atoms. Given protein p, we represent its surface as a connected, non-directed *atom graph* $G^p = (V^p, E^p)$ where each node $v \in V^p$ represents a surface atom of p, and E^p includes all and only the edges (v_i, v_j) such that atoms v_i and v_j are adjacent on the surface of p.

Each node $v \in V^p$ is associated with the vector $\mathbf{x}(v)$ of the 3D coordinates of the center of the corresponding atom and with a local value for each of three features: *curvature*, $cur(v)$, *electrostatic potential*, $elp(v)$, and *hydrophobicity*, $hyd(v)$. Details on how these values are computed and discretized are reported in the following subsections.

4.1.1. Curvature

The local curvature at node v is estimated on the discrete surface formed by its adjacent nodes, thus considering the region within the 1-ring neighborhood of v. According to the method proposed in Ref. [30], we first define a *mean curvature normal operator* as

$$\mathbf{K}(v) = \frac{1}{2A} \sum_{j \in N} (\cot \alpha_j + \cot \beta_j)(\mathbf{x}(v) - \mathbf{x}(v_j)), \tag{1}$$

where

- N is the set of 1-ring neighbor nodes of v;
- A is the local surface area around v, calculated as an extension of the Voronoi area which is still valid even in obtuse triangulations;
- α_j and β_j are the two angles opposite to edge (v, v_j) in the two triangles sharing this edge as in Fig. 2.

The *mean curvature* in v, $K_m(v)$, is then computed as half the magnitude of $\mathbf{K}(v)$ [30].

Finally, we smooth the effects of a local computation of the mean curvature by defining

$$cur(v) = \frac{1}{\sharp N + 2} \left(2K_m(v) + \sum_{j \in N} K_m(v_j) \right), \tag{2}$$

where $\sharp N$ is the cardinality of N.

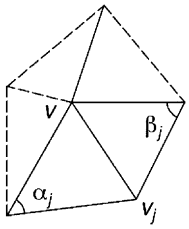

Fig. 2. 1-ring neighbors of v and angles opposite to edge (v, v_j).

4.1.2. Electrostatic potential

It has been demonstrated that the electrostatic potential on protein surface rules ligand approaches and molecular docking [31], in fact biomolecular interactions frequently involve associations between complementary surface potential.

The potential in v, $elp(v)$, is obtained by averaging the local values of potential on the meshes that belong to the solvent-accessible surface of atom v; the electrostatic potential of meshes is computed by the MOLMOL program [27] according to the Poisson–Boltzmann equation [32].

4.1.3. Hydrophobicity

Hydrophobicity is a characteristic of materials that have little or no tendency to absorb water. Conversely, hydrophilicity is a characteristic of materials exhibiting an affinity for water. So, while water is readily absorbed by hydrophilic materials, it tends to form discrete droplets on the surfaces of hydrophobic materials.

There is no agreement on an analytical function that describes how this property is distributed on the surface. We compute the hydrophobicity in atom v, $hyd(v)$, as follows:

$$hyd(v) = h_{aa} \frac{SA_v}{TA_{aa}}, \tag{3}$$

where aa is the amino acid to which atom v belongs, h_{aa} is its hydrophobicity according to the widely used Kyte–Doolittle scale [33], SA_v is the area of the solvent-accessible surface of v, and TA_{aa} is the total area of the surface of aa.

Electrostatic potential and hydrophobicity are not independent. In fact, while protein regions with neutral potential may either be hydrophobic or hydrophilic, positive and negative regions are always hydrophilic. On the other hand, since the interactions ruled by electrostatic potential are much stronger, hydrophobicity turns out to be chemically relevant only for neutral regions.

4.1.4. Feature discretization

Requirement (2) states that patch homogeneity should not be evaluated against the exact values of surface features, but rather against a proper discretization of their range. Three categories are relevant for the curvature: *convex*, *planar*, and *concave*. Three categories are relevant for the potential: *negative*, *neutral*, and *positive*. Finally, only two categories are relevant for hydrophobicity: *hydrophobic* and *hydrophilic*. Since the application domain yields no evidence of a sharp separation between the categories, we define parametric gray areas as depicted in Fig. 3. So, discretization is ruled by five parameters: α_{cur} and β_{cur} are, respectively, the center and the width of the gray area between plain and not plain curvature; α_{elp} and β_{elp} are, respectively, the center and the width of the gray area between neutral and not neutral potential; β_{hyd} is the width of the gray area between hydrophilicity and hydrophobicity. The role of gray areas will be made clear in the following sections.

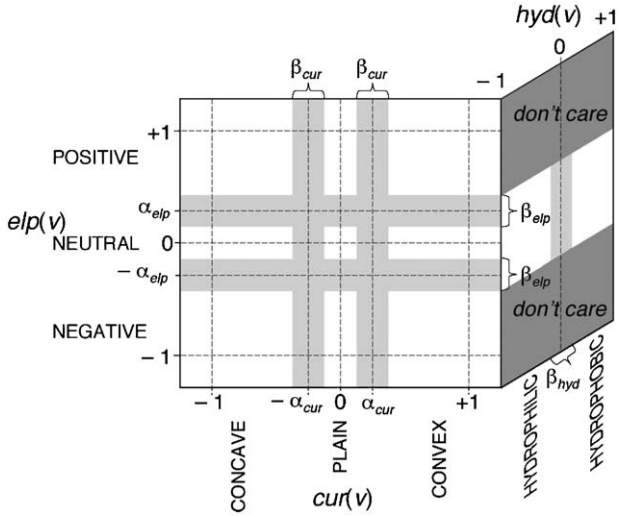

Fig. 3. Discretization for surface features.

Based on this discretization, we define the category(ies) of node v as follows:

$$\hat{cur}(v) = \begin{cases} concave & \text{if } cur(v) < -\alpha_{cur} + \frac{\beta_{cur}}{2}, \\ plain & \text{if } -\alpha_{cur} - \frac{\beta_{cur}}{2} < cur(v) < \alpha_{cur} + \frac{\beta_{cur}}{2}, \\ convex & \text{if } cur(v) > \alpha_{cur} - \frac{\beta_{cur}}{2}, \end{cases} \quad (4)$$

$$\hat{elp}(v) = \begin{cases} negative & \text{if } elp(v) < -\alpha_{elp} + \frac{\beta_{elp}}{2}, \\ neutral & \text{if } -\alpha_{elp} - \frac{\beta_{elp}}{2} < elp(v) < \alpha_{elp} + \frac{\beta_{elp}}{2}, \\ positive & \text{if } elp(v) > \alpha_{elp} - \frac{\beta_{elp}}{2}, \end{cases} \quad (5)$$

$$\hat{hyd}(v) = \begin{cases} hydrophilic & \text{if } hyd(v) < \frac{\beta_{hyd}}{2}, \\ hydrophobic & \text{if } hyd(v) > -\frac{\beta_{hyd}}{2}. \end{cases} \quad (6)$$

The nodes for which a single category is returned are said to be *white*. Those for which a pair of categories are returned, meaning that there is no sharp assignment for them, are said to be *gray*.

4.2. The target functions

As said in Section 1, the domain experts were unable to formulate a priori a single, specific target function to be used for optimizing clustering. On the other hand, they agreed that both homogeneity of features, shape regularity, and large percentage coverage are desirable characteristics for patches. In the following we introduce a quantification for these characteristics, that will be used both to drive the clustering algorithms and to evaluate them:

- *Homogeneity*. We recall that, according to the domain experts, it must be measured with reference to categories rather than exact values. Given a patch c including m nodes, its category according to feature f, $\hat{f}(c)$, is defined as the category of the average value of f for the nodes in c. Note that, as will be explained in Sections 4.3 and 4.4, our clustering algorithms only generate white patches, i.e., patches for which the average value of all features is white. Node v is said to be *compatible* with patch c on feature f iff $\hat{f}(c) \subseteq \hat{f}(v)$, i.e., if either (a) they are both white and their category is the same, or (b) c is white, v is gray, and v falls into a gray area adjacent to the category of c. The homogeneity of c, ranging in $[0, 1]$, is then defined as

$$hom(c) = \begin{cases} \frac{\sharp cur + \sharp elp + \sharp hyd}{3m} & \text{if } \hat{elp}(c) = neutral, \\ \frac{\sharp cur + \sharp elp}{2m} & \text{otherwise,} \end{cases} \quad (7)$$

  where $\sharp f$ is the number of nodes in c that are compatible with c on feature f. The reason why hydrophobicity is considered only for neutral patches has been explained in Section 4.1.3. Given clustering C, its homogeneity $hom(C)$ is the average homogeneity of its patches.
- *Regularity*. The regularity of patch c is computed as the square root of the ratio of the minimum and maximum autovalues in the covariance matrix of the projection of the nodes in c on the plain orthogonal to the normal to c (the inverse of *eccentricity*, [34]). Regularity ranges between 0 (irregular shape) and 1 (circular shape). The regularity of clustering C, $reg(C)$, is the average regularity of its patches.
- *Coverage*. The coverage of clustering C, $cov(C)$, is the percentage of nodes in the protein surface that is assigned to a patch in C.

4.3. Region growing

As seen in Section 2, the region growing approach is widely used in the literature to deal with the problem of surface clustering, due to its undeniable properties of being easy to implement and fast [23]. So, the first technique we propose is based on region growing, properly adapted for managing multiple discrete features, and it builds homogeneous patches with possibly irregular shapes. Let p be a protein and G^p be its atom graph as seen in Section 4.1; the technique consists of three steps:

(1) *Patch initialization*: The initial set of patches, $C = \{c_1, \ldots, c_n\}$, is defined by including all the patches made of adjacent white nodes in $G^p = (V^p, E^p)$ that

```
AssignGreyNodes(G^p, C):
repeat
{  anyAssigned=FALSE;
   for each v_i ∈ V^p not assigned yet do
   {  c_j = getPatch(v_i);
// returns the patch of the nearest assigned node that is
// connected with v_i and compatible with all its features
      if c_j != NULL then
      {  patch[i] = c_j;
         anyAssigned=TRUE;
      }
   }
// actual assignment of v_i is delayed to reduce
// dependency on the ordering of nodes
   for each v_i ∈ V^p not assigned yet do
      if patch[i]! = NULL then assign v_i to patch patch[i];
} until !anyAssigned.
```

Fig. 4. Pseudo-code for the algorithm that assigns gray nodes to patches.

share the same category for curvature, potential, and hydrophobicity. Gray nodes, i.e. nodes whose value falls into a gray area for *at least one feature*, are not assigned at this time.

(2) *Patch growing*: The gray nodes are assigned to one of the existing patches according to the algorithm sketched in Fig. 4. Remarkably, assigning gray nodes to patches according to their distance helps to obtain regularly shaped patches. Besides, since assignment is done through separate iterations, the dependence of the result on the ordering of nodes is reduced.

(3) *Suppression of small patches*: All patches including less than 10 nodes are removed from C, and their nodes are labeled as unassigned.

The time for clustering a single protein including about 500 surface atoms with this region growing technique is absolutely negligible. A qualitative evaluation of the results can be done for instance on Fig. 5a, that shows the clustering obtained for protein 2PVB (PDB code for Parvalbumin).

4.4. Template matching

The main drawback of the region growing technique is the irregular shape of the resulting patches. On the other hand, jointly optimizing the regularity of patches and their homogeneity would very often lead to very small patches, which would be incompatible with requirement (1) concerning patch size. In the light of this, the second clustering technique we experimented is based on template matching, i.e. it looks for the optimal way of assigning nodes to patches whose shape resembles that of a given template. To this end we used a circular template since it is the one that maximizes regularity as defined in Section 4.2.

The technique consists of two steps:

(1) *Definition of candidate patches*: A *circular patch* c with center on node v and integer radius r is a patch consisting of all the nodes whose path distance from v on the atom graph is less than r edges. The set C_{cand} of candidate patches consists of all the white circular patches, centered on all nodes in the atom graph, for which the percentage of compatible nodes on each feature is higher than a threshold γ. Note that several patches centered on the same node may be included in C.

(2) *Optimization*: An optimal set of non-overlapping patches is selected from C_{cand} by exactly solving a pure binary integer programming problem; optimality is defined by encouraging large patches (which may lead to reduce surface coverage). Let t be the number of candidate patches and m_i be the number of nodes in candidate patch c_i; the integer formulation is as follows:

$$\textit{Maximize} \quad \sum_{i=1}^{t} m_i^2 x_i \tag{8}$$

$$\textit{Subject to} \quad x_i \in \{0,1\} \quad \text{for } i = 1,\ldots,t, \tag{9}$$

$$x_i + x_j \leqslant 1 \quad \forall c_i, c_j;\ c_i \cap c_j \neq \emptyset. \tag{10}$$

Each binary variable x_i has value 1 if patch c_i is in the solution, 0 otherwise. Constraint (10) is aimed at avoiding overlapping patches. Note that the homogeneity criterion is not considered in the objective function, since it has already been considered when selecting candidate patches.

The overall size of the optimization problem mainly depends on the extension of the protein and on the value chosen for γ; in the average it includes about 80 variables and 1500 constraints, and it can be solved in less than 0.03 s on a Pentium IV 2.6 GHz equipped with 1 GB of RAM by a software package for linear programming problems (such as ILOG CPLEX) with Mixed Integer Programming option. For a qualitative comparison with region growing, see Fig. 5b that shows the clustering obtained by template matching for protein 2PVB.

5. Tests

The tests we carried out are aimed at evaluating the two proposed algorithms in terms of *optimality*, *robustness*, and *stability*. In principle, two approaches could be followed to this end: a *direct* one, where the clustering algorithms are evaluated by directly measuring the target function, and an *indirect* one, aimed at evaluating the capability of the algorithms to generate clusterings that effectively capture protein similarities. Direct evaluation is hardly feasible in our context, since a single, unquestionable target function is not available (target functions for the single requirements have been defined in Section 4.2, but their relative importance is a subject for discussion). Thus, we resorted to indirect evaluation in the context of the framework for protein classification described in Section 3, where clustering is functional

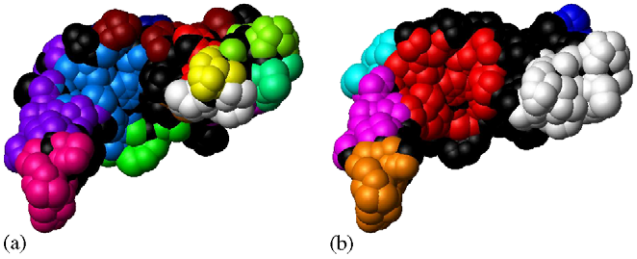

Fig. 5. Clustering of protein 2PVB using region growing (a) and template matching (b) (in black the unassigned areas).

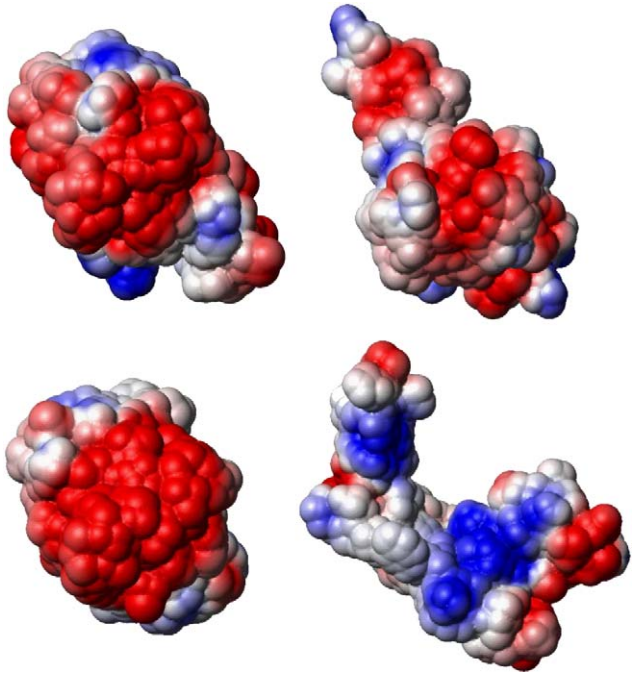

Fig. 6. Electrostatic potential for the fours seeds: 1CLL (calmodulin, top left), 1IRJ (migration inhibitory factor-related protein 14, top right), 2PVB (parvalbumin, bottom left), 1QR0 (4′-phosphopantetheinyl transferase Sfp, bottom right). Blue, red, and white respectively mean positive, negative, and neutral potential.

to surface comparison and to the mining of frequent surface patterns.

In particular, we used as input for the tests a set of *mutation chains* generated by homology modeling [35]. Starting from a seed protein in the Protein Data Bank, each mutant model is obtained from the previous one by introducing five punctual mutations at a time in a small surface area. Adopting this ad hoc dataset guarantees a fine control over the surface properties, hence over surface similarity, which is particularly good for evaluating optimality and robustness. Besides, this dataset enables us to verify whether clustering properly captures surface similarities by analyzing how the classification obtained reflects the mutation chains. Note that none of the existing protein classifications in the literature could have been used to our purposes. In fact, structural classifications—such as SCOP [36] and CATH [37]—are based on concepts that are different from, and often in contrast with, those we consider. On the other hand, the other approaches to surface-based classification either consider geometrical features only [6], or are focused on a single surface patch (already recognized to play a key role for a certain function) that covers a limited percentage of the surface, thus ignoring the overall surface properties [10].

5.1. The dataset

The dataset we adopted was obtained starting from four different proteins (*seeds*, see Fig. 6), that were subjected to

Table 1
Seed descriptions

Protein	Superfamily	Family	# Mutations
1CLL	EF-hand	Calmodulin-like	12
1IRJ	EF-hand	S100	13
2PVB	EF-hand	Parvalbumin	11
1QR0	Phosphopantetheinyl transferase	Phosphopantetheinyl transferase SFP	13

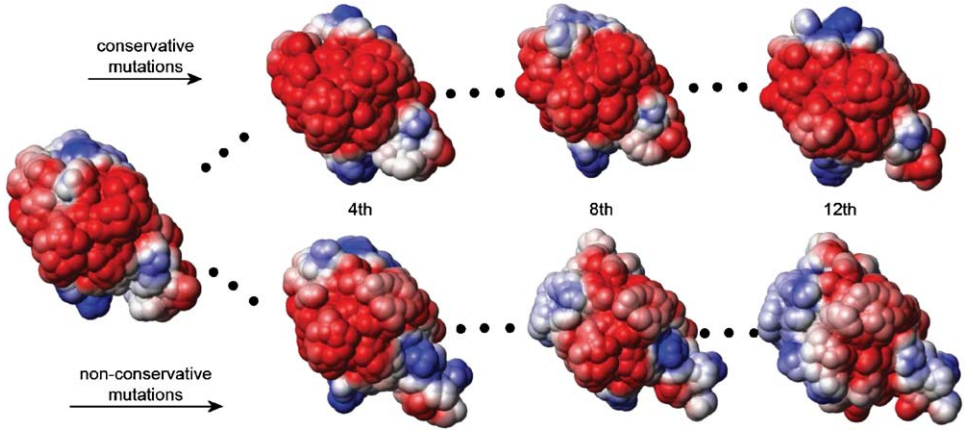

Fig. 7. Electrostatic potential for six mutations of seed 1CLL.

progressive in silico mutations to obtain 98 protein models that are more and more different from the generating seeds. The characteristics of the seeds are reported in Table 1: all of them have similar size and three out of four belong to the same SCOP superfamily. Mutations consist in replacing five amino acids that are neighboring on the surface at a time, until all the surface amino acids have been replaced. They are obtained by the *homology modeling technique* [35], that provides models with stable configurations as validated by PROSA II [38]. Note that, though the proteins we used for testing are not representative—in terms of structure—of the whole set of known proteins (34201 in the Protein Data Bank on 13 December 2005), they are nonetheless significant since mutations cover the whole range of values of each surface feature we consider.

More precisely, each seed generates one *group* of mutant protein models distributed along two *chains*: one is generated through *conservative mutations*, that impact on the shape but preserve the chemical surface properties (e.g. replacing aspartic acid with glutamic acid, both negatively charged); the other chain is generated through *non-conservative mutations*, that also alter the chemical properties (e.g. replacing aspartic acid, that is negatively charged, with threonine, that is neutral). The number of mutants in each chain ranges between 11 and 13, depending on the seed backbone. We define the *distance* between two proteins of the same group as the number of mutations that separate them. Fig. 7 shows the electrostatic potential distribution on the surface for 6 out of 24 mutations for seed 1CLL. Note that, from the structural point of view, mutants are always related since surface mutations limitedly affect the overall structure. Conversely, the impact on surface properties (in particular for non-conservative mutations) is much more relevant, thus only the close mutants are strictly related from the biochemical point of view.

As to intra-chain similarity, we note that surface similarity progressively decreases within each chain when the distance increases. Fig. 8a shows, for the two chains generated from seed 1IRJ, how the percentage of matching atoms depends on the distance from the seed.[1] Fig. 8b shows which percentage of the matching atoms also maintain a similar electrostatic potential (a difference lower than 0.1 V). As expected, the curve for non-conservative mutations is steeper.

Finally, in order to tune parameters for mining and classification, we organized proteins in two datasets, DS1 and DS2, the first one including the proteins generated from seeds 1CLL and 1IRJ, the second one those from seeds 2PVB and 1QR0. Tests were then carried out by using in turn each dataset as the training set and the other as the test set.

5.2. Optimality

Optimality is related to the capability of the algorithms to produce clusters that meet the requirements expressed in terms of homogeneity, regularity, and coverage and that lead to sound classifications in the context of the overall approach described in Section 3.

The soundness of classification is affected by *inter-group* and *intra-group* errors [12]. Inter-group errors occur when proteins generated from different seeds are included in the same class. Intra-group errors occur when a class includes non-consecutive mutants of the same group. Of course, inter-group errors are much more serious since proteins in different groups are less similar than those in the same group.

All tests returned no inter-group errors, so we do not define any measure for them in this paper. Conversely, in order to measure intra-group errors for a classification $S = \{s_1, \ldots, s_g\}$ into g classes, where each class is a set of proteins of the same group, we define its *scattering* as

$$\frac{1}{g}\sum_i \frac{dist_i + 1 - \sharp s_i}{\sharp s_i}, \tag{11}$$

where $dist_i$ is the maximum distance between two proteins of class s_i. Fig. 9 shows the scattering for datasets DS1 and DS2 and for the two algorithms at the different depths of the classification dendrogram; at the far left, the finest classification (all singleton classes), at the far right, the coarsest one (two classes).[2] Some comments on these results:

- The absence of inter-group errors is an encouraging outcome, even if the dataset includes four classes, since three seeds belong to the same SCOP super-family thus sharing a similar folding, which makes it more difficult to distinguish them. However, we argue that a more relevant result is the low number of intra-group errors. Even in presence of very similar mutants, the classification obtained closely follows the mutation chains, thus proving that the clustering algorithms correctly capture surface similarities.
- The scattering is higher in presence of large classes, while it is lower when the classes are very small or when they are a few.
- The scattering is zero when only two classes are created since, as said above, both clustering algorithms lead to classifications that do not present any inter-group error. Thus, the two classes created by the coarsest classification exactly match the two groups of each dataset.
- The region growing algorithm outperforms the template matching one for both datasets.
- Both algorithms give better results when the similarity between proteins is higher, in fact 58% of intra-group

[1] Two atoms in two proteins are considered to *match* if they have the same name and the positions of their amino acids within the polypeptidic sequence are the same.

[2] All tests reported in this section are based on the following setting for parameters: $\alpha_{elp} = 0.275$, $\beta_{elp} = 0.05$, $\alpha_{cur} = 0.09$, $\beta_{cur} = 0.02$, $\beta_{hyd} = 0.0001$, $\gamma = 0.75$.

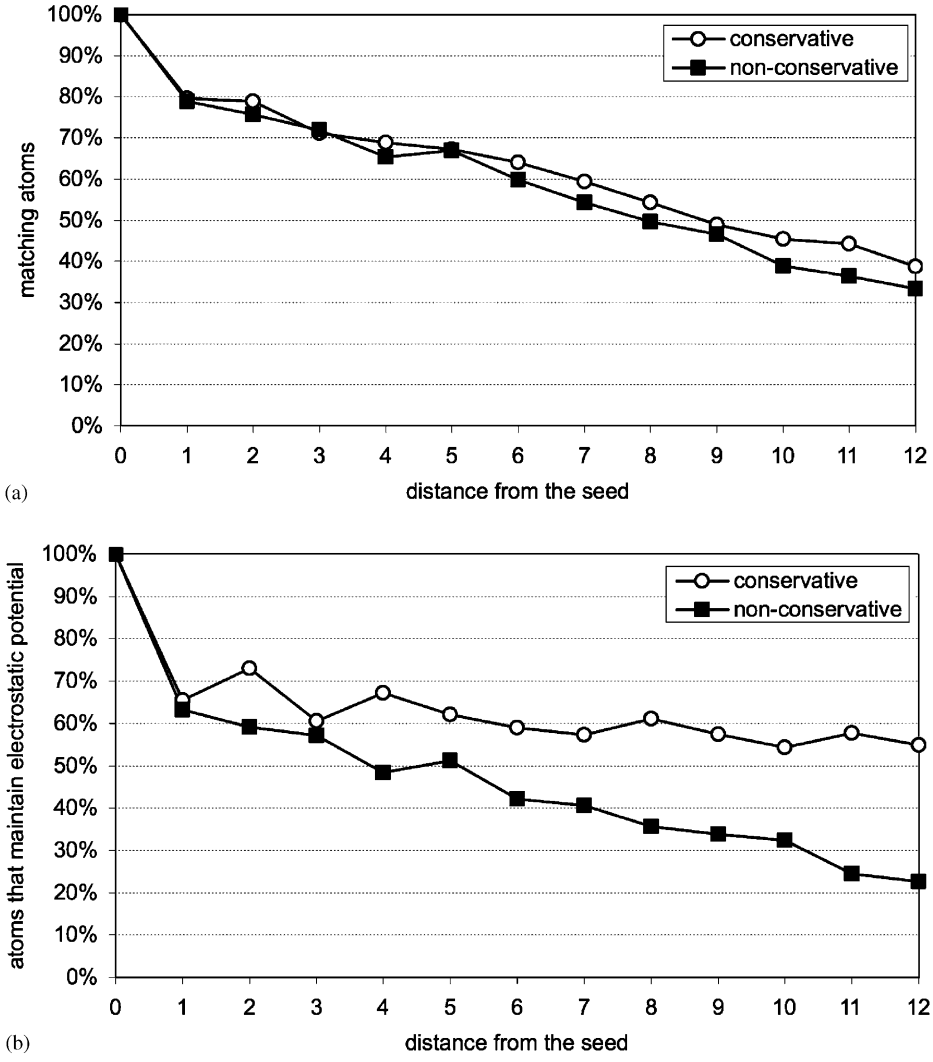

Fig. 8. For mutants generated from seed 1IRJ: (a) percentage of matching atoms; (b) percentage of matching atoms that maintain electrostatic potential.

errors are made on non-conservative chains. The reason why DS1 yields more errors is that proteins are more similar to each other since the two seeds belong to the same superfamily.

Table 2 reports the average values of the target functions for the two datasets. As expected, homogeneity is maximum for region growing while regularity is higher for template matching. The results also confirm the obvious conflict between regularity and coverage.

5.3. Robustness

Robustness is related to the capability of the algorithms to cluster similar proteins in the same way. This property preserves the approach from failing when the protein surfaces are affected by noise or present small differences that should not impact on clustering. In particular, a protein may temporarily modify its shape and properties while interacting with another one. As for optimality, an indirect confirmation of robustness is given by the results of classification: in fact, the ability to correctly classify similar proteins in the same class depends on the ability to generate similar clusterings for similar surfaces.

In order to further measure the robustness we extend the Jaccard coefficient [39]. Let two proteins be given, and let a surface clustering be defined on each of them. As already said, some of the surface atoms of the two proteins may match; based on this matching we define *SS* as the number of couples of atoms that in both clusterings are clustered together, and *SD* as the number of couples of atoms that are clustered together in one clustering but not in the other. Then we compute a Jaccard coefficient[3] for the two clusterings as

$$JAC = \frac{SS}{SS + SD}. \tag{12}$$

[3] The Jaccard coefficient should not be confused with the percentage of atoms belonging to homologous patch in the two clusterings.

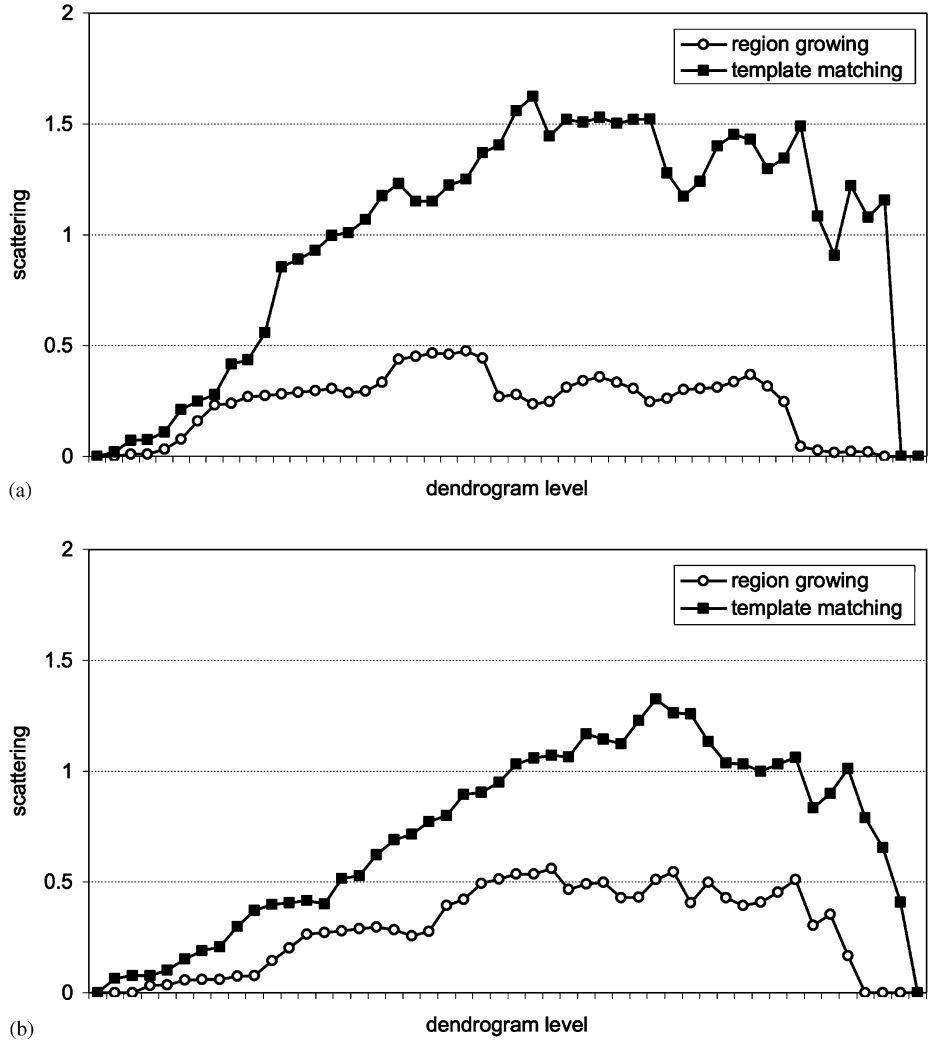

Fig. 9. Scattering for different dendrogram depths on DS1 (a) and DS2 (b).

Table 2
Average value of target functions

		avg $hom(C)$	avg $reg(C)$	avg cov(C)
DS1	Region growing	1	0.49	0.55
	Template matching	0.85	0.74	0.44
DS2	Region growing	1	0.45	0.58
	Template matching	0.85	0.71	0.46

Table 3
Average Jaccard coefficient for all couples of proteins with increasing distance (number of mutations)

	Distance = 1	Distance = 2	Distance = 3
Region growing	0.29	0.28	0.27
Template matching	0.32	0.32	0.31

Note that, when computing *JAC*, a dummy patch containing all the unassigned atoms in each clustering is added. This enables a better characterization of similarity since it is desirable that also the unassigned atoms are the same in similar clusterings.

Table 3 reports the average Jaccard coefficient for all the couples of proteins in the whole dataset separated by 1, 2, and 3 mutations. Both algorithms exhibit a sufficient degree of robustness. Region growing is slightly less robust than template matching since the strong relevance given to homogeneity makes it more sensitive to small alterations in the surface.

5.4. Stability

Stability is related to the capability of the algorithms to produce similar clusterings when run with similar parameters, and is desirable since it reduces the impact of

Table 4
Average Jaccard coefficient in function of the parameter values

		α_{elp}	α_{cur}	β_{elp}	β_{cur}	γ
5%	Region growing	0.72	0.55	0.91	0.84	—
	Template matching	0.78	0.40	0.95	0.67	0.59
10%	Region growing	0.56	0.35	0.84	0.71	—
	Template matching	0.66	0.27	0.90	0.54	0.44

parameter tuning on the clustering results. A measure of stability can be obtained by computing the Jaccard coefficient on the same protein clustered with two different parameter sets. For each parameter, except β_{hyd} whose value does not significantly impact on clustering, we varied its value starting from the optimal one used in Section 5.2 so as to change the feature category for 5% and 10% of the atoms in the dataset. For example, changing α_{elp} from 0.275 to 0.315 makes 5% more of the atoms neutral.

Table 4 shows that the algorithms are more sensitive to some parameters than to others. In particular, the decay induced by changes in the curvature parameters is faster than that related to potential. This problem is intrinsic to the application domain since, while curvature continuously changes across the surface, potential is stable in large areas. In other words, the atoms whose curvature category changes are widely scattered on the protein surface, which generates noise that negatively impacts on clustering. Conversely, the atoms whose potential categories change, are concentrated in a few areas that will remain homogeneous, though in a different category.

6. Comparison and conclusions

In this paper we discussed the issues related to clustering of protein surfaces, framed within an approach to surface-based protein classification. After illustrating the requirements for clustering, we proposed two techniques mainly aimed at creating homogeneous and regular surface patches, respectively.

We close the paper by making an overall comparison of the two techniques. Both display good capability of identifying significant patches, as confirmed by the classification results obtained from the optimality test. The region growing technique appears to outperform template matching since it always produces less intra-group errors; it is not clear whether the errors in template matching are mainly due to insufficient homogeneity of patches or to insufficient coverage of the surface. Note that the different relevance given to homogeneity also affects the number of patches created: on average, region growing and template matching produce, respectively, 13 and 8 patches for each protein. A qualitative evaluation of the difference between the clusterings obtained with the two techniques can be done by comparing Figs. 5a and b. On the other hand, a quantitative evaluation can be expressed by the Jaccard coefficient computed on two clusterings made on the same protein with the two techniques: the average coefficient for the whole dataset is 0.24, which confirms that there is a marked difference between region growing and template matching. In particular, from the above we may infer that protein surfaces are better characterized by strongly homogenous patches, even if their shape is irregular; shape regularity should be given low relevance since it induces a reduction in the portion of surface covered by patches.

In conclusion, the results obtained suggest that multi-feature clustering of protein surfaces is expressive enough to enable significant comparison of proteins. We are currently working to give an interpretation of the patches obtained and to exploit the surface patterns in different bioinformatics applications, in particular classification. As specifically concerns the clustering algorithms, our future work will be aimed at improving the effectiveness of template matching by relaxing the shape regularity constraint; three possible solutions we are considering consist in adopting elliptical templates, in letting the patches partially overlap, and in adopting the patches obtained by template matching as seeds for a region growing method.

Acknowledgements

We would like to thank Francesco Capozzi and Maria Turano, from the Department of Food Science of the University of Bologna (Italy), for their precious support in identifying the crucial requirements for surface clustering in the biochemical domain, in preparing the mutation chains, and in validating the clustering and classification results.

References

[1] I. Shindyalov, P. Bourne, Protein structure alignment by incremental combinatorial extension of the optimal path, Protein Eng. 11 (1998) 739–747.

[2] N. Alexandrov, D. Fischer, Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures, Proteins 25 (1996) 354–365.

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

[4] S. Pickering, A. Bulpitt, N. Efford, N. Gold, D. Westhead, AI-based algorithms for protein surface comparisons, Comput. Chem. 26 (2001) 79–84.
[5] D. Fischer, H. Wolfson, S. Lin, R. Nussinov, Three-dimensional, sequence order-independent structural comparison of a serine protease against the crystallographic database reveals active site similarities, Protein Sci. 3 (1994) 769–778.
[6] M.A. Lozano, F. Escolano, Protein classification by matching and clustering surface graphs, Pattern Recognition 39 (4) (2006) 499–736.
[7] Y. Kaneta, et al., A method of comparing protein molecular surface based on normal vectors with attributes and its application to function identification, in: Proceedings of the Joint Conference on Information Science, 2002.
[8] M. Teodoro, G. Phillips Jr., L. Kavraki, Molecular docking: a problem with thousands of degrees of freedom, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2001, pp. 960–966.
[9] K. Kinoshita, J. Furui, H. Nakamura, Identification of protein functions from a molecular surface database, ef-site, Struct. Funct. Genomics 2 (1) (2002) 9–22.
[10] F. Ferrè, et al., SURFACE: a database of protein surface regions for functional annotation, Nucl. Acid Res. 32 (2004) 240–244.
[11] X. Wang, Finding patterns on protein surfaces: algorithms and applications to protein classification, IEEE Trans. Knowl. Data Eng. 17 (8) (2005) 1065–1078.
[12] L. Baldacci, M. Golfarelli, Mining complex patterns from molecular surfaces, in: Proceedings of the International Workshop on Biological Data Management, 2005.
[13] L. Kauvar, H. Villar, Deciphering cryptic similarities in protein binding sites, Curr. Opin. Biotechnol. 9 (4) (1998) 390–394.
[14] M. Connolly, Solvent-accessible surface of proteins and nucleic acids, Science 221 (1983) 709–713.
[15] C. Hofbauer, H. Lohninger, A. Aszódi, Surfcomp: a novel graph-based approach to molecular surface comparison, Chem. Inf. Comput. Sci. 44 (2004) 837–847.
[16] T. Seidl, H. Kriegel, A 3D molecular surface representation supporting neighborhood queries, in: Proceedings of the International Symposium on Large Spatial Databases, 1995, pp. 240–258.
[17] C. Schillo, G. Herrmann, F. Ackermann, S. Posch, G. Sagerer, Statistical classification and segmentation of biomolecular surfaces, in: Proceedings of the International Conference on Image Processing, 1995, pp. 560–563.
[18] F. Cazals, F. Chazal, T. Lewiner, Molecular shape analysis based upon the Morse–Smale complex and the Connolly function, in: Proceedings of the ACM Symposium on Computational Geometry, 2003, pp. 237–246.
[19] D. Luebke, A developer's survey of polygonal simplification algorithms, IEEE Comput. Graph. Appl. 21 (3) (2001) 24–35.
[20] E. Zuckerberger, A. Tal, S. Shlafman, Polyhedral surface decomposition with applications, Comput. Graph. 26 (5) (2002) 733–743.
[21] A. Mangan, R. Whitaker, Partitioning 3D surface meshes using watershed segmentation, IEEE Trans. Vis. Comput. Graph. 5 (4) (1999) 308–321.
[22] K. Wu, M. Levine, 3D part segmentation using simulated electrical charge distributions, IEEE Trans. Pattern Anal. Mach. Intell. 19 (11) (1997) 1223–1235.
[23] R. Adams, L. Bischof, Seeded region growing, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 641–647.
[24] G. Lavoué, F. Dupont, A. Baskurt, Constant curvature region decomposition of 3D-mesh by a mixed approach vertex-triangle, in: Proceedings of the WSCG, 2004, pp. 245–252.
[25] T. Srinark, C. Kambhamettu, A novel method for 3D surface mesh segmentation, in: Proceedings of the International Conference on Computer, Graphics, and Imaging, 2003.
[26] Y. Zhang, J. Paik, A. Koschan, M. Abidi, A simple and efficient algorithm for part decomposition of 3-d triangulated models based on curvature analysis, in: Proceedings of the International Conference on Image Processing, 2002.
[27] R. Koradi, M. Billeter, K. Wüthrich, MOLMOL: a program for display and analysis of macromolecular structures, Mol. Graph. 14 (1996) 51–55.
[28] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, P. Bourne, The protein data bank, Nucl. Acids Res. 28 (1) (2000) 235–242.
[29] A.K. Jain, et al., Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[30] M. Meyer, M. Desbrun, P. Schroeder, A. Barr, Discrete differential geometry operators for triangulated 2-manifolds, in: Proceedings of the Visualization and Mathematics, 2002.
[31] B. Honig, A. Nicholls, Classical electrostatics in biology and chemistry, Science 268 (1995) 1144–1149.
[32] A. Nicholls, B. Honig, A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson–Boltzmann equation, J. Comput. Chem. 12 (1990) 435–445.
[33] J. Kyte, R. Doolittle, A simple method for displaying the hydropathic character of a protein, Mol. Biol. 157 (1) (1982) 105–132.
[34] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1993.
[35] F. Eisenmenger, P. Argos, R. Abagyan, A method to configure protein side-chains from the main-chain trace in homology modelling, Mol. Biol. 231 (3) (1993) 849–860.
[36] L. Lo Conte, B. Ailey, T. Hubbard, S. Brenner, A. Murzin, C. Chothia, SCOP: a structural classification of proteins database, Nucl. Acids Res. 28 (1) (2000) 257–259.
[37] F. Pearl, et al., The CATH database: an extended protein family resource for structural and functional genomics, Nucl. Acids Res. 31 (1) (2003) 452–455.
[38] M. Sippl, Recognition of errors in three-dimensional structures of proteins, Proteins: Struct. Funct. Genet. 17 (1993) 355–362.
[39] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Cluster validity methods: Part I, SIGMOD Rec. 31 (2) (2002) 40–45.