Characterizing the Morphology of Protein Binding Patches

ARTICLE *in* PROTEINS STRUCTURE FUNCTION AND BIOINFORMATICS · DECEMBER 2012

Impact Factor: 2.63 · DOI: 10.1002/prot.24144 · Source: PubMed

3 AUTHORS, INCLUDING:

Noël Malod-Dognin
Imperial College London
24 PUBLICATIONS **92** CITATIONS
SEE PROFILE

Frederic Cazals
National Institute for Research in Compute…
103 PUBLICATIONS **1,435** CITATIONS
SEE PROFILE

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Characterizing the Morphology of Protein Binding Patches

Noël Malod-Dognin — Achin Bansal — Frédéric Cazals

N° 7743

Septembre 2011

_ Computational Biology and Bioinformatics _

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7743--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche SOPHIA ANTIPOLIS - MÉDITERRANÉE

Characterizing the Morphology of Protein Binding Patches

Noël Malod-Dognin , Achin Bansal , Frédéric Cazals

Theme : Computational Biology and Bioinformatics
Computational Sciences for Biology, Medicine and the Environment
Équipes-Projets Algorithms Biology Structure

Rapport de recherche n° 7743 — Septembre 2011 — 45 pages

Abstract: Understanding the specificity of protein interactions is a central question in structural biology, whence the importance of models for protein binding patches—a patch refers to the collection of atoms of a given partner accounting for the interaction. To improve our understanding of the relationship between the structure of binding patches and the biological function of protein complexes, we present a binding patch model decoupling the topological and geometric properties. While the geometry is classically encoded by the 3D positions of the atoms, the topology is recorded in a graph encoding the relative position of concentric shells partitioning the interface atoms. The topological - geometric duality provides the basis of a generic dynamic programming based algorithm to compare patches, which is instantiated to respectively favor topological or geometric comparisons.

On the biological side, using a dataset of 92 co-crystallized structures organized in biological sub-families, we exploit our encoding and the two comparison algorithms in two directions. First, we show that Nature enjoyed the topological and geometric degrees of freedom independently while retaining a finite set of qualitatively distinct topological signatures, and show that topological similarity is a less stringent notion that the ubiquitously used geometric similarity.

Second, we analyze the topological and geometric coherence of binding patches within sub-families and across the whole database, and show that complexes related to the same biological function can encompass geometrically distinct shapes.

Previous work on binding patches focused on the investigation of correlations between structural parameters and biochemical properties on the one hand, and on structural comparison algorithms on the other hand. We believe that the abstraction coded by the topological - geometric paves the way to new classifications, in particular in the context of flexible docking.

Key-words: Protein interface, 3D-structure, comparison, maximum clique, tree-edit-distance.

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Caractérisation de la morphologie des patchs de liaison protéiques

Résumé : Comprendre la spécificité des interactions protéiques est une question centrale en biologie structurale, d'ou l'importance de modèles pour les patchs de liaisons protéiques - un patch étant l'ensemble des atomes d'un partenaire donné qui participent à l'interaction. Pour améliorer notre compréhension de la relation entre les structures des patchs de liaisons protéiques et les fonctions biologiques des complexes, nous présentons un modèle de patchs qui découple les propriétés topologiques et géométriques. Tandis que la géométrie est encodée de manière classique par les coordonnées 3D des atomes, la topologie est enregistrée dans un graphe encodant les positions relatives de couches concentriques qui partitionnent les atomes de l'interface. La dualité entre la topologie et la géométrie est la base d'un algorithme de comparaison de patchs, reposant sur une programmation dynamique générique qui est instanciée pour favoriser soit une comparaison topologique, soit une comparaison géométrique.

Du point de vue biologique, en utilisant une base de donnée de 92 structures de complexe co-cristalisés organisés en sous-familles biologiques, nous exploitons notre encodage et nos algorithmes de comparaisons dans deux directions. Premièrement, nous montrons que la nature utilise les degrés de libertés topologiques et géométriques indépendamment, tout en ne conservant qu'un ensemble fini de signatures topologiques qualitativement distinctes, et nous montrons que la similarité topologique est une notion moins stricte que la similarité géométrique habituellement utilisée.

Deuxièmement, nous analysons la cohérence topologique et géométrique des patchs de liaisons à l'intérieur des sous-familles et sur la base de donnée complète, et nous montrons que les complexes impliqués dans les mêmes fonctions biologiques possèdent des formes géométriquement distinctes.

Les travaux précédents sur les patchs de liaisons protéiques se concentraient d'une part sur la recherche de corrélations entre des paramètres structuraux et des propriétés biochimiques, et d'autre part sur des algorithmes de comparaisons structurales. Nous croyons que l'abstraction codée par la topologie et la géométrie ouvre la voie vers de nouvelles classifications, en particulier dans le contexte de l'amarrage (docking) flexible.

Mots-clés : Interface protéique, structure 3D, comparaison, clique maximum, distance d'édition d'arbre

Contents

1 Introduction

Biology rests on macro-molecular complexes, so that a central question consists of understanding the determinants of the stability and the specificity of binding. These questions have been approached from two complementary perspectives, namely experiments and modeling. On the experimental side, structures resolved by X ray crystallography and NMR are of fundamental importance, as they lay the ground for modeling studies [JBC08], but also pave the way to protein engineering [FWE$^+$11]. Structural information is also complemented by directed mutagenesis and binding affinity measurements, which convey information of biological and thermodynamical nature [Fer99], and by evolutionary information [LW10b]. Structural modeling work on the other hand, aims at developing explanatory and predictive models, and may be classified into two veins. To describe them, the following terminology is used to describe a binary complex: a *binding patch* refers to a collection of atoms on one partner, responsible for the interaction; the union of two such patches defines the *interface* of the complex.

Dissecting the morphology of interfaces and patches. The design of explanatory and predictive models rests on the identification of structural parameters (geometric and topological) which best describe the biological and biophysical properties of interfaces [JBC08]. The first task when studying an interface is to identify the atoms contributing to the two binding patches, as the buried surface area of these atoms in the complex often reliably hints at the stability of the interaction [CCJ99]. To study the amino-acid composition and more generally the biochemical properties of patches, the *core-rim* model was introduced based on the accessibility of atoms in the complex [CJ02]. This model was also used to show that conserved residues tend to locate in the core [GC05]. In a more biological perspective, double mutant cycles were used to evidence the modular structure of binding patches [RRA$^+$05]. On the prediction side, algorithms computing putative patches on a molecular suface have been developed. Their strategy consists of generating patches according to specific model, and the putative patches are assessed against those observed in co-crystallized complexes. In [JT97] and [ASP$^+$09], the patch model consists of picking the k-nearest exposed neighbors of a central atom, resulting in disk-shaped patches.

Comparing and clustering patches. The second vein is concerned with the design of tools performing structural comparisons and alignments, with two privileged applications. The first one is the analysis of bound complexes and the classification of their interfaces. In [KTWN04], a classification of all interfaces of the PDB results into 103 classes split into three groups, based on the structural similarities inferred using geometric hashing, and the folds of the subunits. In a nearby vein, the SCOPPI database classifies patches using successive criteria related to the SCOP domains, structural similarities, and sequence identity [WHKS06]. The second one is the detection of similar binding patches on two unbound partners, a problem reminiscent from docking. While similarity detection is at the very heart of docking [HMWN02], or particular interest are the methods inferring similar patches from the proteins' exposed surface. In particular, in [KJ10], similar patches are inferred by merging small graphs (graphlets)

whose $RMSD_d$ is upper-bounded, and which exhibit comparable bio-chemical properties.

Contributions. As just recalled, several geometric models have been proposed to investigate specific biological - biophysical properties. On the other hand, we showed in [CPBJ06] that most if not all these parameters could be retrieved from a single geometric construction [Caz10], and we also demonstrated in [BGNC09] that sharper conclusions could be derived at the single complex level—as opposed to the database level, for the the study of correlations between structural parameters versus the the conservation, the polarity, as well as the solvation properties of residues. This work takes our Voronoi modeling of interfaces and patches one step further, by proposing a hierarchical patch model which can be seen as a parameterized core-rim model. We abstract this model into a graph so as to decouple the topological and geometric properties of patches, and present a detailed study on a database of 92 co-crystallized heterodimers.

With respect to dissection studies,we compare the nature of the geometric and topological design patters used by Nature, and also study the symmetry of interactions across interfaces. With respect to comparison and clustering studies, we present a versatile atomic level matching algorithm, favoring either topological or geometric features. In fact, our encoding is used to solve a relaxed structural alignment problem through a maximum clique calculation—a NP-hard problem. Intuitively, our graph encoding corresponds to a partitioning of a patch into so-called *shells*, which is especially useful to run structural alignments by searching quasi-isometric subsets, as initially proposed in [BW87]. Assuming that two patches are given as lists of atoms BP_1 and BP_2, we seek the largest subsets $A_1 = \{a_i\}_{i=1,\dots,n}$ and $A_2 = \{b_i\}_{i=1,\dots,n}$ of these lists, together with a one-to-one correspondence $a_i \leftrightarrow b_i$ between them, such that the RMSD between internal distances is bounded by a user defined threshold ε, that is

$$RMSD_d(BP_1, BP_2) = \sqrt{\frac{2}{n \times (n-1)} \sum_{i<j} |d_{a_i,a_j} - d_{b_i,b_j}|^2} \leq \varepsilon. \qquad (1)$$

This problem has long been known to be equivalent to the maximum clique problem in the so-called product graphs [BW87, MDAY10]. The maximum clique is a well known NP-Hard problem [Kar72] that can be tackled with enumeration algorithms [BK73, CK08] or optimization algorithms [BBPP99, Ö02, MDAY10]. Given that the size of the product graphs is quadratic in the number of atoms, and that typical interfaces involve from 100 to 500 atoms, the product graphs have a number of vertices beyond tens of thousands (up to 133216 vertices with our benchmark), which is intractable for exact algorithms. To fudge around this difficulty, we use the aforementioned graph to *localize* the application of the maximal clique algorithms to shells. As we shall see, the problems solved still challenges the state-of-the-art maximum clique algorithms.

2 Theory

We first present our encoding of patches as graphs, the associated comparison algorithms, and the application to database analysis.

2.1 A Hierarchical Encoding of Patches

Outline. Consider a binary complex, and assume that the *interface* atoms have been identified on both partners. Focusing on a given partner in the Solvent Accessible model, where the radii have been expanded by $r_w = 1.4$, we define its *binding patch*, called patch for the sake of conciseness, as the solvent accessible surface (SAS) of its interface atoms. A patch therefore consists of spherical polygons called *faces*, with two neighboring faces intersecting along a circle arc. The peripheral atoms of such a patch make up its rim, and to measure the distance of a face to the patch boundary, each face is assigned an integer called its *shelling order* (SO). Finally, the SO are used to decompose the patch into *concentric shells*, whose relative positions are encoded in a graph called the *face shelling tree*, from which another graph called the *atom shelling tree* is derived. These notions are respectively illustrated and explained on Figs. 3 and 4(a,b). We now sketch the main steps of the atom shelling tree construction, and refer the reader to the supplemental section 5.1 for the details.

Defining patches. We identify the interface atoms with our Voronoi interface model [CPBJ06, BGNC09], whose definition and software are presented in [LC10] and [Caz10]. Let A and B be the two species of the complex, also called partners or subunits, and denote W the water molecules squeezed in-between the partners. Let the *restriction* of a ball B_i be the 3D region defined by the intersection between B_i and its Voronoi region. (To be precise, the Voronoi region refers to the region of the ball in the power diagram of the balls of the SAS model). Two atoms are called *neighbors* provided that their restrictions intersect. A water molecule is called *interfacial* provided that its has neighbors on both partners. An *interface atom* is an atom which is neighbor to the other partner's atoms, or to interfacial water molecules.

Having identified the interface atoms, we process the two subunits separately. The *patch* of a subunit is defined as the SAS of this partner *restricted* to its interface atoms. The patch therefore consists of spherical polygons called *faces*; a face is bounded by circle arcs, and two faces are called *incident* if they share a circle arc; a circle arc is itself bounded by the points found at the intersection of three spheres. The patch is encoded in a data structure giving access to the faces and the circle-arcs, and their incidences.

Face Shelling Tree. Shelling consists of assigning an integer value called *shelling order* or SO to each face, and is best presented in terms of graph distance. Term a face a *boundary face* if one of its bounding circle arcs is incident to one interface atom and one non-interface atom. Such faces are assigned a SO of zero. Consider now the dual graph of the patch: the nodes of this graph are the faces; two nodes are connected by an edge provided that the associated faces share a circle-arc. The *shelling order* or SO of a face is its shortest distance to a boundary face in the dual graph. Note that the contribution of an atom to the patch may consist of several faces with different SO. The SO and the dual graph are used to compute the following topological encoding. First, we define a *shell* as a maximal connected component of the dual graph involving faces with the same shelling order, so that the patch is partitioned into shells. Second, we encode the relative position of shells within the so-called *face shelling tree*. This

tree contains one node $N_{SG}(s)$ for each shell s, the *node size* being the number of faces. To see how the edges of the shelling tree are defined, consider two incident faces whose SO differ of one unit. Let s and t be the shells containing these faces, and assume that $SO(s) > SO(t)$: the face shelling tree contains one arc from $N_{SG}(s)$ to $N_{SG}(t)$. The number of outgoing arcs of a node is called its *arity*.

Atom Shelling Tree. In order to base the comparison of patches on atoms rather than faces, we edit the face shelling tree into an *atom shelling tree*. The process consists of substituting atoms to faces, with the following special cases: if an atom is present several times in the same shell, it is counted once; if an atom belongs to several shells in a branch of the face shelling tree, it is assigned to the shell closest to the root of the tree. Finally, the sons of a node are sorted by increasing size i.e. number of atoms, resulting in an *ordered* atom shelling tree, called shelling tree for short in the sequel.

Note that the atom shelling tree encodes topological information namely the relative position of the shells, while the 3D coordinates of the atoms within the shells encode the geometry.

2.2 Comparing Patches: a Generic Dynamic Programming based Approach

Encoding a patch as an ordered tree whose nodes contain shells paves the way to patch comparison using dynamic programming [Bil05]. More precisely, to compare two trees, we edit one into the other, computing the so-called Tree Edit Distance (TED). The TED, whose details are recalled in the supplemental section 5.2.1, is based on three operations, namely node deletion, node insertion, and node morphing. The TED calculation delivers an Ordered Edit Distance Mapping, namely a set $M \subset Vertices(T_1) \times Vertices(T_2)$ such that for any pair $(v_1, v_2) \in M$ and $(w_1, w_2) \in M$, one has: (i) $v_1 = w_1$ iff $v_2 = w_2$, or (ii) v_1 is an ancestor of w_1 iff v_2 is an ancestor of w_2, or (iii) or v_1 is to the left of w_1 iff v_2 is to the left of w_2. (Recall that trees are ordered.)

In our case, given that a node corresponds to a shell of atoms, adjusting the substitution cost yields strategies to compare the topology and the geometry of patches.

Topological comparison. To identify common patterns of nested shells encoded within the shelling tree, we compute the TED with the following costs. Adding or deleting a node associated with a shell s has a cost of $|s|$, namely the number of atoms in the shell. Morphing a shell s_1 into a shell s_2 corresponds to matching $\min(|\ s_1\ |, |\ s_2\ |)$ atoms in-between the two shells, or equivalently, to a cost $\max(|\ s_1\ |, |\ s_2\ |) - \min(|\ s_1\ |, |\ s_2\ |)$. At the patch level, the atoms matched by the TED calculation are called *isotopologic* since they belong to nodes of the shelling trees satisfying the constraints (i,ii,iii) of the edit distance mapping. Denoting $\mathrm{SIM}_t(T_1, T_2)$ the number of isotopologic atoms between the BP, the TED cost is the following symmetric difference

$$\mathrm{TED}_t(T_1, T_2) = |\ T_1\ | + |\ T_2\ | - 2\ \mathrm{SIM}_t(T_1, T_2). \quad (2)$$

This number being upper-bounded by $|T_1|+|T_2|$, it yields the dissimilarity score $\in [0,1]$:

$$\mathrm{DIS}_t(T_1,T_2) = \mathrm{TED}_t(T_1,T_2)/(\mid T_1 \mid + \mid T_2 \mid), \tag{3}$$

which can be interpreted as the percentage of non-common atoms.

Geometric comparison. Consider now the problem of comparing the geometry of BP, as specified by Eq. (1). Because a brute-force attempt to solve this problem for the whole patch is intractable, we restrict the identification of quasi-isometric subsets to pairs of shells. That is, we define a second TED calculation as follows.

As previously, the cost of inserting/deleting a shell s is $\mid s \mid$. For the morphing cost between shells s_1 and s_2, assume that $\mid s_1 \bigcap s_2 \mid$ quasi-isometric atoms have been identified by a maximum clique calculation, as specified by Eq. (1) (see details in the supplemental section 5.2.2). The morphing cost is equal to the size of their symmetric difference, namely $\mid s_1 \mid + \mid s_2 \mid -2 \mid s_1 \bigcap s_2 \mid$. Denote $\mathrm{SIM}_g(T_1,T_2)$ the number of atoms matched across all pairs of nodes in the edit distance mapping. The corresponding tree edit distance counts the number of un-matched atoms, namely:

$$\mathrm{TED}_g(T_1,T_2) = \mid T_1 \mid + \mid T_2 \mid -2\ \mathrm{SIM}_g(T_1,T_2). \tag{4}$$

Mimicking Eq. (3), we define:

$$\mathrm{DIS}_g(T_1,T_2) = \mathrm{TED}_g(T_1,T_2)/(\mid T_1 \mid + \mid T_2 \mid) \tag{5}$$

Topology versus geometry. The previous two criteria both rely on a TED calculation, and report isotopologic atoms. Yet, the geometric comparison is more stringent, and $\mathrm{SIM}_g(T_1,T_2) \leq \mathrm{SIM}_t(T_1,T_2)$. Equivalently, the topological dissimilarity is a lower bound of the geometric dissimilarity, that is $\mathrm{DIS}_g(T_1,T_2) \geq \mathrm{DIS}_t(T_1,T_2)$.

2.3 Database Analysis

A dataset of n co-crystallized protein complexes yields a database $\mathcal{P}$ of $2n$ patches. We assume that the database is organized into biological families corresponding to biological functions. We further split each biological family by distinguishing the ligand and the receptor of each complex. Thus, the database of patches is partitioned into *typed bio-families*. For example, the typed bio-family AA_Pept_R refers to the patches of receptors (i.e. immunoglobulins) from the bio-family of antibody-antigen complexes having peptides as antigens. This decomposition scheme aims at performing structural comparisons in conjunction with the analysis of biological functions. Prosaically, we wish to investigate whether it makes sense to speak of the patches of say immunoglobulin - peptides complexes.

Denoting P the patches of a typed bio-family, let $\overline{P}$ be the set of patches that are the partners of the ones in P, and P^c be the set of patches neither in P nor in $\overline{P}$. Note that $\mathcal{P} = P \cup \overline{P} \cup P^c$. For a given patch p, define $P_{\setminus p}$ such that $P = \{p\} \cup P_{\setminus p}$. We shall use the following partition of the database induced by any patch to test hypothesis on the similarity of patches:

$$\mathcal{P} = p \cup P_{\setminus p} \cup \overline{P} \cup P^c. \tag{6}$$

Given a patch p and a dissimilarity score $s(p,q)$, practically that of Eq. (3) or Eq. (5), we denote by $\hat{p}$ the nearest neighbor i.e. the patch of the database with lowest dissimilarity:

$$\hat{p} = \arg \min_{q \in \mathcal{P} \setminus \{p\}} s(p,q). \tag{7}$$

We distinguish the following cases: **case I:** $\hat{p} \in P_{\setminus p}$, **case II:** $\hat{p} \in \overline{P}$, and **case III:** $\hat{p} \in P^c$. Case I directly measures the compatibility between the dissimilarity score and the typed bio-family classification, and the possibility of using of the dissimilarity score for generating automatic classifications of patches. Case II is related to the symmetry (or lack of) between partner patches across an interface, a feature especially interesting for heterodimers—homodimers are symmetrical, albeit not always exactly. Case III highlights contradictions between the typed bio-family classification and the dissimilarity score values.

3 Results

The results presented in this section were obtained on a database containing 184 patches generated from the dataset of $n = 92$ complexes presented in section 5.3. These complexes are heterodimers, as homodimers are usually symmetric, and one of our goals is to study the symmetry of patches.

The 184 shelling trees have the following characteristics: from 26 to 271 atoms, from 3 to 14 shells, a SO in the ranging from 2 to 7, and an arity from 1 to 10. Also, the 184 patches yield a total of $\binom{184}{2} + 184 = 17020$ pairwise comparisons.

3.1 Morphological Studies: Topology versus Geometry

Topological signatures. To analyze the morphology of patches beyond the core-rim model, we first assess the repartition of atoms in a patch by plotting the number of atoms against the number of shells, see Fig. 5. Since this plot exhibits a continuous variation, we extract typical morphologies by examining extreme cases for a fixed number of shells and atoms, respectively. As illustrated on Fig. 6, minimizing and maximizing the number of atoms for a fixed number of shells yields *tubular* and *pyramidal* shapes. Similarly, minimizing and maximizing the number of atoms for a fixed number of atoms yields *anisotropic* and *isotropic* shapes, respectively.

To understand the specificity of these shapes, we plot for each patch the variation of the number of atoms as a function of the SO. Inspection of all curves (supplemental Figs. 1 to 5) allows us to single out the cases of Fig. 7. The most frequent pattern is that of a decreasing curve, which corresponds to a large rim followed by nested shells of smaller size. This happens for the case for the pyramidal, isotropic and anisotropic cases—the latter case being characterized by a smaller number of shells. The tubular shape is characterized by an almost horizontal curve, corresponding to a patch with constant diameter. Aside from these monotonic curves, one also observes increasing-decreasing curves, corresponding to the *pear-like* morphology, illustrated on Fig. 6.

Having singled out these shapes, we compare the two patches of a complex resorting to the *average shelling order* or $\overline{SO}$ of a patch, defined as the sum of the shelling order of each atom of the patch divided by the number of atoms.

This quantity is maximized for a linear tree implying that most of the atoms are deep inside the patch, and is minimized for *flat* trees, implying that most of the atoms are located around the patch border. For a given complex, the asymmetry of its patches are witnessed by different $\overline{SO}$, and the two families AA_Pept and the AA_Prot appear as very asymmetric (supplemental Figs. 6 and Fig. 7). However, while partners patches tend to be asymmetric, similar receptor patches still have similar ligand patches (supplemental Fig. 8).

Geometry versus topology. As observed in section 2.2, the topological and geometric dissimilarities satisfy $\mathrm{DIS}_g \geq \mathrm{DIS}_t$. This property is observed on Fig. 8, which also exhibits patches having a similar topology (low DIS_t) but different geometries (high DIS_g). As an illustration, two such patches having respectively 82% and 32% of common atoms from the topological and geometric standpoints, are presented on the supplemental Fig. 9.

3.2 Morphology based Clustering and Biological Functions

Asymmetry of patches. Table 1 summarizes the identification results for TED_t and TED_g, with different thresholds for the latter. We first observe that case I is the most frequent whatever the method, with identification rates much higher than for case II—ratio of $47.2/11.4 = 4.14$ for TED_t, and up to $73.4/3.8 = 19.31$ for TED_g with $\epsilon = 2$. The fact that the nearest neighbor of a patch p typically lies in the bio-family P of p rather than in $\overline{P}$ witnesses the asymmetry of the patches of a complex. To statistically backup this observation, we computed for each typed bio-family P the set of all pairwise scores $s(P,P), s(P,\hat{P})$ and $s(P,P^c)$ using the score of TED_g at $\epsilon = 2$Å. Comparing the distributions of $s(P,P)$, $s(P,\overline{P})$ and $s(P,P^c)$ using the Wilcoxon-Mann-Whitney rank test yields p-values smaller than 7.25×10^{-4}, confirming the asymmetry (supplemental Table 5).

The identification rate culminates at about 73% for TED_g with a distance threshold ϵ of 2, an observation which also holds at the typed bio-family level (supplemental table 3), and which also holds for complexes whose resolution lies in the range 2 to 3, highlighting the robustness of the method (supplemental Table 4). To better understand the efficacy of TED_g in the context of typed bio-families and biological functions, we computed the number of correct identifications restricted to the patches such that $\mathrm{DIS}_g(p,\hat{p}) \leq t$, varying t in the range $[0,1]$. As seen from Fig. 9, all the identifications are correct for $\mathrm{DIS}_g(p,\hat{p}) \leq 0.24$, but the number of erroneous identifications monotonically increases beyond $t = 0.24$. Incorrectly identified instances highlight inconsistencies in the original organization of the database, in the sense that patches which are obviously different are found in the same typed bio-family. Two such examples are presented on the supplemental Fig. 11.

Analysis of biological families. We finally use our dissimilarity scores to cluster the patches. Given a dissimilarity score $s(p,q)$ and a dissimilarity threshold δ, consider the graph $G_\delta = (V,E)$, whose vertices V are the patches, with an edge between two patches p and q provided that $s(p,q) \leq \delta$. For a fixed threshold, we define the clusters as the connected components of G_δ. Studying the evolution of the number of clusters when varying δ provides another way of assessing the consistency/homogeneity of a typed bio-family.

Using the topological score with threshold δ_t, for all bio-families but AA_Pept_L, the patches cluster at $\delta_t = 0.15$, showing that the typed bio-families are consistent from the topological standpoint (supplemental Fig. 10(left)). On the other hand, using the geometric score, a threshold of $\delta_g = 0.55$ is required to obtain the same coalescence—an observation consistent with the lack of patches with low dissimilarity score as seen on Fig. 9. These relative values of δ_t and δ_t evidence the stringency of a geometric versus a topological comparison.

The correctness of the identifications performed with $\epsilon_g \leq 0.24$ suggests using this threshold to cluster the correctly identified patches, in order to compare the clusters and the typed bio-families (supplemental Table 6). While the paucity of data makes the inference of general claims difficult, it is interestingly noticed that some original typed bio-families remain grouped, while others get dispatched across the clusters generated. The former class illustrates groups of geometrically similar patches involved in a given biological function.

4 Discussion

Morphological studies. The question of understanding which features of patches account for the specificity of biological interactions has been examined in two veins. On the one hand, a number of works performed correlation studies between structural parameters (interface size, planarity, modularity, organization into a core and a rim), and biophysical properties (composition, conservation, solvation, $\Delta\Delta G$) [CCJ99, CJ02, BCRJ03, BCRJ04, GC05]. Selected such parameters have also been traced along molecular dynamics [ML07]. While trends have emerged for collections of complexes [ML07, GC05, JBC08], conclusions from meta-analysis, when refined under the lens of sharper structural parameters, may not apply to isolated complexes [BGNC09]. On the other hand, the search of binding sites on orphan proteins motivated the development of patch models, which when instantiated on a protein surface, allow the comparison between these instantiations and the patches observed in a co-crystallized complex [JT96, JT97, ASP$^+$09]. So far, the patch models proposed are isotropic ones, as they consist of tracing a geodesic disks on the molecular surfaces.

In this context, this work ellaborates on our Voronoi interface model [CPBJ06, BGNC09, LC10], which offers a unified way to refine classical interface parameters [JBC08], and proved instrumental to transpose conclusions from the database level to the single complex level, regarding the biochemical properties of interfaces [CJ02], the geometry of conservation [GC05], as well as the solvation of residues along molecular dynamics trajectories [ML07].

The atom shelling tree construction extends the shelling of Voronoi interfaces, and brings novelties in the two directions. With respect to correlation studies, the atom shelling tree is a hierarchical encoding of the patch, replacing a binary attribute (location of an atom in the rim or the core) by an integer-valued one (the atom shelling order). The shelling tree makes it possible to study the topology of a patch—a dimension ignored so far, independently from its geometry. While a continuous distribution of patches is observed with respect to topological and geometric features, we have shown that typical patch morphologies, namely tubular, pear-like, pyramidal, isotropic and anisotropic, could be singled out. Our encoding also allows comparing two patches of a com-

plex, either directly from the dissimilarity score, or indirectly based on statistics derived from the atom shelling tree—the average shelling order $\overline{SO}$.

Three developments should prove particularly useful. The first one is related to the *inverse problem*. The atom shelling trees are generated from the observed patches of complexes, but the number biological complexes known is very limited compared to the number of known unbound protein structures and crystal contacts. The inverse problem would consist in finding the optimal mapping of a given atom shelling tree onto a given protein surface. Because, as illustrated by our canonical morphologies, a patch cannot be reduced to an isotropic shape, a solution to this problem would enhance the search of putative patches on orphan proteins, in the spirit of [JT97, ASP$^+$09]. The second one is related to the asymmetry detection and to partner retrieval. The asymmetry reported in this work is related to the non-flatness of interfaces. This hinders the possibility to retrieve the possible partners of a given patch, in particular for docking applications. Technically, the asymmetry detection comes from the fact that our generic dynamic programming based matching algorithms produces and Ordered Edit Distance Mapping, which is a one-to-one atom mapping. Developing a more general, say k-to-k atom mapping would allow adjusting the level of non-symmetry tolerated as a function of the parameter k. This extension poses challenging graph-theoretical problems, but would prove useful for docking, in combination with filters avoiding steric clashes and forcing the biochemical compatibility between the atoms matched. Finally, our topological comparison should also prove useful to assess docking results, as done in the CAPRI experiment. While current assessment are based on geometric criteria [LW10a], comparing predicted and crystallized patches, but also crystallized patches and their pre-image on the unbound protein, should be useful in the context of flexible docking to understand deformation modes.

Morphology based clustering and Biological Functions. The problem of comparing and clustering interfaces and patches motivated work in two directions. On the one hand, approaches have been developed for co-crystallized complexes. In [KTWN04, KN07], interfaces are clustered both from the patch and the whole structure point of view, using geometric hashing techniques applied to the C_α carbons. Identical interfaces are assigned to so-called class I or class II clusters depending on whether they involve chains with similar or different fold, while class III clusters regroup similar patches—the patches are similar but the interfaces are not. In a nearby vein, the SCOPPI database [WHKS06] classifies patches using a two step approach. First, domain sequences from the same SCOP family are aligned, and then all patches are mapped over their aligned sequence. This 0-1 vector called Interface Tag (IFT) is used as the signature of the patch—0 codes a non-interface residue while 1 codes an interface residue. Patches are then clustered into the same family if the cosine angle distance between their two IFT is larger than 0.8. (We note in passing that the IFT comparison does not take into account the gaps induced by the multiple sequence alignments, and thus does not convey any information on the coverage of interface atoms, as opposed to our dissimilarity score.) Second, the obtained families are clustered using selected geometric criteria. On the other hand, tools have been developed to compare solvent accessible patches. In Probis [KJ10], graphlets encoding the proximity of functional groups are first defined. Selected

graphlets are compared using a maximum clique approach, and the global match between two patches is obtained by merging graphlets.

Our matching algorithms depart from these works in two major ways. First, we perform matching at the atomic level, as opposed to the residue and functional group levels in [KTWN04] and [WHKS06]. Second, we accommodate independently topological and geometric comparisons. We have shown that the former is more lenient, which also explains why typed bio-families, which correspond to biological functions, are topologically but not geometrically consistent. Interestingly, the geometric comparison of patches at a threshold where all identifications are correct allows to exhibit patches involved in the same function but with different geometry.

These specificities call for further developments. In the clustering of interfaces and patches of [KTWN04, KN07], class III clusters gather geometrically similar patches involved in dissimilar interfaces, i.e. patches having more than one binding function. As patches can be similar at the residue level but dissimilar at the atomic level, our comparison tools should be useful to refine such analysis. In a different spirit, our ability to handle coherently topological and geometric criteria calls for the development of hierarchical classification of patches, based on a combination of topological, geometric and biological (sequence) information, in a manner similar the classification of quaternary structures performed in [LPLCT06]. A solution to the aforementioned inverse problem would allow performing such studies not only on the biological complexes of the PDB, but also on isolated proteins of known structure. In a related spirit, we envision applications of such analysis to patches from the immune system, in the context of the IMGT_3D database, see `http://www.imgt.org/`. Such studies would be particularly meaningful in the context of the *collier de perles* annotations [RL02], which assigns a unique numbering to the amino-acids of the complementarity determining regions, which are responsible for the specificity of the immune response in general and for affinity maturation in particular.

5 Material and Methods

5.1 A Hierarchical Encoding of Patches

In this section, we present the details of the atom shelling tree construction, sketched in section 2.1, and illustrated on Fig. 4.

5.1.1 Defining Patches

The identification of interface atoms is carried out by seeking edges present in the α-complex of the expanded balls, and whose endpoints belong to the two partners or involve an interfacial water molecule, see [CPBJ06, BGNC09, LC10]. Practically, the α-shape is computed using the *Alpha_shape_3* package of the Computational Geometry Algorithms Library, see `www.cgal.org`. The underlying algorithm has randomized complexity $O(n \log n + k)$, with n the number of input balls and k the size of the output—the number of simplices of the regular triangulation underlying the α-shape.

This done, the complex is dissociated, and another 0-complex is computed for each partner, in the Solvent Accessible Surface (SAS) model. On a per-partner basis, this 0-complex is used to compute a combinatorial representation of the boundary of the expanded atomic balls [AE96], stored in a half-edge data structure (HDS) [dBvKOS97]. The HDS consists of faces, half-edges, and vertices, and the connectivity information between these items allows in particular to find the connected component of the patch, and to find the cycles bounding a given connected component. For example, a patch consisting of a surface patch with a hole in the middle is bounded by two cycles, each consisting of a consecutive circle-arcs. In the sequel, such cycles are called Connected Components of the Boundary, or CCB.

From a geometric standpoint, a robust 3D embedding of the HDS is obtained using exact degree two algebraic numbers to represent the coordinates of the point lying at the intersection of three spheres [CCLT09].

5.1.2 Selected Properties of Patches

Before presenting the details of the topological encoding sketched in section 2.1, we discuss selected properties of patches.

Patches with multiple rims. In all generality, a connected component of the patch is not simply connected i.e. it may contain holes. This is illustrated on Fig. 4(c,d), where the packing defect in the middle of the interface is such that the patch is topologically equivalent to an annulus. To understand the implications of this fact on the shelling process, consider a connected patch with several CCB, and assume that the faces incident on the half-edges of these CCB have been initialized to one. Computing the shells as described in section 2.1 would result in a directed acyclic graph (DAG) rather than a tree, with a number of roots equal to the number of CCB.

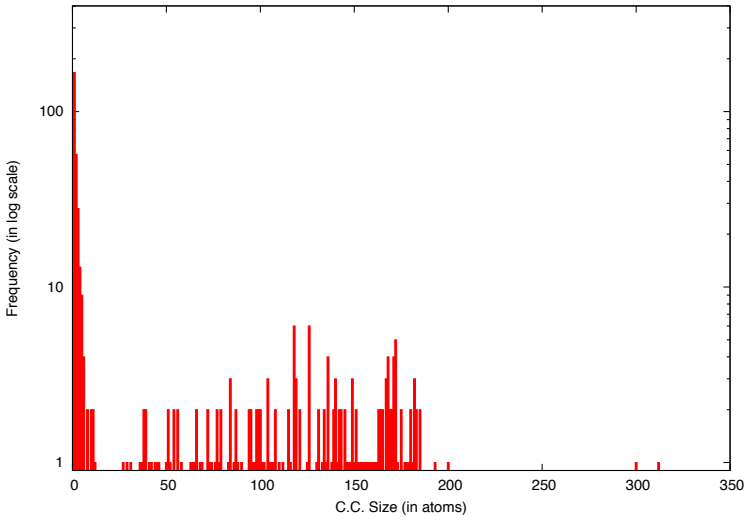
To fudge around this difficulty, the outer cycle only is used to initialize the SO calculation by tagging selected faces with a SO of zero. In doing so, the resulting DAG is a tree.

Patches with multiple connected components. Since a patch is defined as the SAS of the interface atoms of a sub-unit, it can be disconnected. This happens if the contact between partners has two distinct regions, which typically occurs for large protein interfaces [BCRJ04]. But as illustrated on Fig. 4(a,b), this also happens due to packing defects within one subunit. In this case, the shelling graph may contain several small connected components (cc).

However, one expects one component to contain significantly more atoms than the remaining ones. Processing all patches and plotting the histogram of the size of the atom shelling trees yields Fig. 1, which has an empty gap between the range [1,12] and [27,312].

Practically, having computed the connected component of the patch, we remove all components containing less than 12 faces— results in the removal of at most 12 atoms. In all cases processed, we are left with a unique cc, which is the largest one. This fact accounts for the name *face shelling tree* as opposed to *face shelling forest*.

Figure 1 Frequencies of connected component of given size. Computed over the 184 shelling graphs generated from our dataset, the histogram presents an empty gap between the range [1,12] and [27,312].



On patches and atom selected. A final comment is in order to qualify the atoms which are involved in our patch construction, in particular with respect to interface model based on the loss of solvent accessibility — $\Delta ASA > 0$. As observed in [CPBJ06] and explained in [Caz10], some interface atoms selected by the Voronoi model may no lose solvent accessibility. This happens in particular for atoms which are buried in their sub-unit: such atoms are interface atoms in the Voronoi model, but are excluded from the patch model since they do not contribute to the SAS of the sub-unit. Such cases represent less than 10% of all interface atoms [CPBJ06].

5.1.3 Algorithm

We are now ready to detail the algorithm sketched in section 2.1.

Step 1: Computing the HDS. The half-edge data structure encodes the boundary of the union of balls, as computed in [AE96]. A certified embedding in 3D is obtained thanks to the robust geometric operations described in [CCLT09].

Step 2: Computing the Connected Component of the Boundary (CCB). The CCB are the cycles bounding the patches. Given the HDS, finding all CCB of a patch requires running a Union-Find algorithm [Tar83], which has (almost) linear complexity.

Step 3: Computing the Connected Component of Half-edges (CC). To identify the connected components of a patch, we run a Union-Find algorithm on all the half-edges of the patch. Note that each c.c. will yield a shelling graph/tree.

Step 4: Initializing the Shelling Order. From steps 2 and 3, the largest CCB of each connected component is selected, and the corresponding faces are assigned a SO of zero. This step settles the case of connected component with several rims.

Step 5: Computing the Shelling Order. Using the connectivity of faces encoded in the HDS, a priority queue is used to assign the SO to all the faces. The queue is initialized with the boundary faces identified at step 4.

Step 6: Computing the shells. A shell being a connected component of faces having the same S0, a Union-Find algorithm is also called to create the shells.

Step 7: Computing the Face Shelling Graph. A parent-child relationship between two shells is witnessed by a half-edge incident on two faces having a SO which differs by one unit. Collecting all such pairs requires a linear pass over all half-edges. Constructing the Face Shelling Graph from the parent-child list is then straightforward.

Step 8: Selecting the Face Shelling Tree. So far, one tree has been computed for each connected component of the patch. We select the tree selected corresponds to the largest component in the Face Shelling Graph. As discussed above, this settles the case of patch with several connected components.

Step 9: Computing the Atom Shelling Tree from the Face Shelling Tree. Editing the atom shelling tree from the face shelling tree just requires handling atoms contributing several faces to the patch, as discussed in section 2.1.

Step 10: Ordering Atom Shelling Tree. This step requires sorting the sons of a node by increasing size.

5.2 Comparing Patches: a Generic Dynamic Programming based Approach

5.2.1 The Tree Edit Distance

The generic TED. Given two ordered trees T_1 and T_2, i.e. trees such that the children of each node are ordered, the *Tree Edit Distance* calculation aims at *editing* or *morphing* one tree into the other [Bil05]. The TED computation is actually based on three operations, namely deleting a node, inserting a node, and morphing a node of the first tree into a node of the second tree. The output of the TED consists of an *ordered edit distance mapping*, namely a set $M \subset Vertices(T_1) \times Vertices(T_2)$ such that for any pair $(v_1, v_2) \in M$ and $(w_1, w_2) \in M$, one has: (i) $v_1 = w_1$ iff $v_2 = w_2$, or (ii) v_1 is an ancestor of w_1 iff v_2 is an ancestor of w_2, or (iii) or v_1 is to the left of w_1 iff v_2 is to the left of w_2. (Recall that trees are ordered.) Call a node of a tree a *paired node* provided that it is involved in a morphing operation, and let N_1 (resp. N_2) the nodes of T_1 (resp. T_2) which are not paired, and let λ be the empty node. If $\gamma()$ refers to the cost of an insert/delete/morph operation, the cost of the edit distance mapping M is the following:

$$\gamma(M) = \sum_{(v,w)\in M} \gamma(v \to w) + \sum_{v \in N_1} \gamma(v \to \lambda) + \sum_{w \in N_2} \gamma(\lambda \to w) \tag{8}$$

From which one defines the TED as:

$$TED = \min_{M:\text{Edit Distance Mapping}} \gamma(M). \tag{9}$$

It can be shown that the TED calculation is amenable to a dynamic programming approach, which we sketch the sake of completeness.

The recursive structure of the TED. Computing the TED actually requires handling ordered forests rather than trees—one tree for each son of the root. From now on, we consider two forest F_1 and F_2. Denoting v and w the rightmost (if any) roots of F_1 and F_2, and $F_1(v)$ the sub-tree rooted at v—and likewise for F_2. It can be shown that the TED calculation has the following recursive structure:

$$TED = \begin{cases} \delta(\emptyset, \emptyset) & = 0 \\ \delta(F_1, \emptyset) & = \delta(F_1 - v, \emptyset) + \gamma(v \to \lambda) \\ \delta(\emptyset, F_2) & = \delta(\emptyset, F_2 - w) + \gamma(\lambda \to w) \\ \delta(F_1, F_2) & = \min \begin{cases} \delta(F_1 - v, F_2) + \gamma(v \to \lambda) \\ \delta(F_1, F_2 - w) + \gamma(\lambda \to w) \\ \delta(F_1(v), F_2(w)) + \delta(F_1 - F_1(v), F_2 - F_2(w)) + \gamma(v \to w) \end{cases} \end{cases} \tag{10}$$

These equations show that:

- the value of $\delta(F_1, F_2)$ depends on a constant number of problems of smaller size;
- each sub-problem can be computed in constant time.

The optimal algorithm developed in [DMRW07] has cubic time complexity, and quadratic memory requirements. Note that the TED problem for unordered trees is in general NP-hard [Bil05].

Instantiation in the context of Atom Shelling Trees. The three operations insert/delete/morph are generic in the sense that they depend on the semantics associated to the nodes. In our case, a node of the shelling tree corresponds to a set of atoms, and different interpretations can be used, as we have seen in section 2.2: while focusing solely on the number of common atoms yields a topological comparison, namely algorithm TED_t, focusing on quasi-isometric subsets of atoms yields a geometric comparison, namely algorithm TED_g. In the next section, we explain how the geometric comparison of two shells is carried out, which is underlying the morph operation of TED_g.

5.2.2 Comparing Shells

Our strategy to compare shells is a modification of the one proposed in [MDAY10] in the context of protein's alpha-carbon backbone comparison, and reduces to a maximum clique calculation. To present it, we shall need the following definitions and notations.

Definition 1 *A $m \times n$ **2D graph** $G = (V, E)$ is a graph in which the vertex set V is depicted by a (m-rows) $\times$ (n-columns) array T, where each cell $T[i][k]$ contains at most one vertex $i.k$ from V (note that for both arrays and vertices, the first index stands for the row number, and the second for the column number). Two vertices $i.k$ and $j.l$ can be connected by an edge $(i.k, j.l) \in E$ only if $i \neq j$ and $k \neq l$.*

Definition 2 *A **clique** of a graph $G = (V, E)$ is a subset of its vertex set V such that any two vertices in it are adjacent (i.e. connected by an edge in E).*

Definition 3 *The **maximum clique problem** (also called maximum cardinality clique problem) is to find a largest, in terms of vertices, clique of an arbitrary undirected graph G.*

In matching two shells $s_1 \subset BP_1$ and $s_2 \subset BP_2$, the goal is to find a one-to-one correspondance between two sets of atoms $m_1 \subseteq s_1$ and $m_2 \subseteq m_2$. Since we aim, following Eq. (1) at finding quasi-isometric subsets, we shall use constraints. Assume that we wish to match atom $i \in s_1$ with atom $k \in s_2$, and similarly atom $j \in s_1$ with atom $l \in s_2$, and let $d^1_{i.j}$ ($d^2_{k,l}$) be the distance between atoms i and j (resp. k and l). The compatibility constraints between the two pairs go as follows:

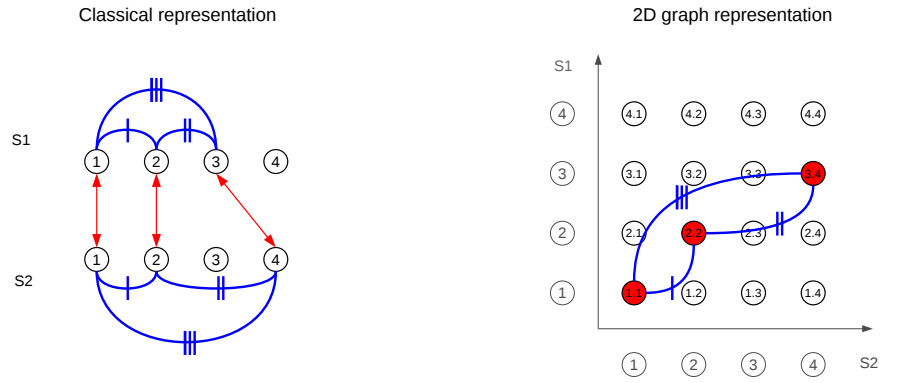
1. $i \neq j$ and $k \neq l$; this constraints ensures that an atom of s_1 can be matched with at most one atom of s_2, and vice versa;
2. for a distance threshold ϵ, we impose $|d^1_{i.j} - d^2_{k.l}| \leq \epsilon$; this constraints ensure that the $RMSD_d$ of internal distances is upper-bounded by ϵ.

A feasible matching is thus a sequence of matching pairs $i_1 \leftrightarrow k_1$,$i_2 \leftrightarrow k_2$,..., $i_n \leftrightarrow k_n$ such that any two pairs are compatible. Searching the largest feasible

matching can be rephrased in a $|s_1| \times |s_2|$ 2D graph $G = (V, E)$ in the following way. Each row i of V represents an atom $i \in s_1$, and each column k represents an atom $k \in s_2$. For all possible matching pairs $i \leftrightarrow k$, we create a vertex $i.k \in V$, on row i, column k. For all compatible couples of matching pairs $i \leftrightarrow k$ and $j \leftrightarrow l$, we create an edge $(i.k, j.l) \in E$. A feasible matching corresponds to a clique in G, and the longest feasible matching to a maximum clique in G. See Fig. 2 for an illustration.

Comparing two shells is modeled as finding a maximum clique in a graph. The maximum clique problem is one of the first problem shown to be NP-Complete [Kar72], and it has been studied extensively in literature. Interested readers can refer to [BBPP99] for a detailed state of the art about the maximum clique problem. In our current implementation, the maximum clique in the 2D graph is computed using the Cliquer library [Ö02].

Figure 2 Classical versus 2D graph representation of feasible matching computation. Left: The red arrows correspond to the feasible matching $1 \leftrightarrow 1, 2 \leftrightarrow 2, 3 \leftrightarrow 4$, which implies that both $d^1_{1.2} \simeq d^2_{1.2}$, $d^1_{1.3} \simeq d^2_{1.4}$, and $d^1_{2.3} \simeq d^2_{2.4}$. Right: the same matching is represented in a 2D-graph.



5.3 Dataset of Biological Complexes

Our approach is validated on a dataset of 92 high resolution ($\leq$ 2Å) protein complexes consisting of 77 antibody/antigen complexes extracted from the IMGT_3D database (`http://www.imgt.org/3Dstructure-DB/`) and of 15 protease/inhibitor complexes coming from [CMJW03]. These 92 complexes yield a database of 184 patches, two per complex. By distinguishing the type (receptor, ligand) of each partner, and using biological information on each complex, these patches are classified into 10 so-called *typed bio-families*, each such group being uniquely identified by a triple "FAMILY_SUBFAMILY_TYPE"; see the supplemental table 2.

5.4 Algorithms TED$_g$: parameters

As specified by Eq. (1), the algorithm TED$_g$ involves a distance threshold ϵ. More precisely, recall that matching atom $i \in s_1$ with atom $k \in s_2$, and atom

$k \in s_1$ with atom $l \in s_2$ requires $|d_{i.j} - d_{k.l}| \leq \epsilon$. The parameter ϵ affects both the quality of the comparisons and their computation times. Quality-wise, the larger ϵ, the larger the internal distance discrepancies allowed, whence the larger and the less similar the common subsets of atoms returned. Computationally, the larger ϵ, the more difficult the identification of quasi-isometric subsets. On computers with Intel Xeon processors at 2.66Ghz, computing the 17020 pairwise comparisons of our database was done in about 575 seconds by TED_t, in about 5832 seconds by TED_g ($\epsilon = 1$Å), and 32419 seconds by TED_g ($\epsilon = 2$Å). In this study, we compared TED_t against TED_g with $\epsilon = 2$Å. As shown while discussing the identification rates—Table 1, $\epsilon = 2$ strikes a balance between the quality of the comparison and the hardness of the computations.

References

[AE96] N. Akkiraju and H. Edelsbrunner. Triangulating the surface of a molecule. *Discrete Applied Mathematics*, 71(1):5–22, 1996.

[ASP+09] L-P. Albou, B. Schwarz, O. Poch, J-M. Wurtz, and D. Moras. Defining and characterizing protein surface using alpha shapes. *Proteins*, 76:1–12, 2009.

[BBPP99] I.M. Bomze, M. Budinich, P.M. Pardalos, and M. Pelillo. The maximum clique problem. *Handbook of Combinatorial Optimization.*, 1999.

[BCRJ03] R.P. Bahadur, P. Chakrabarti, F. Rodier, and J. Janin. Dissecting subunit interfaces in homodimeric proteins. *Proteins: Structure, Function, and Bioinformatics*, 53(3):708–719, 2003.

[BCRJ04] R.P. Bahadur, P. Chakrabarti, F. Rodier, and J. Janin. A dissection of specific and non-specific protein-protein interfaces. *J. Mol. Biol.*, 336, 2004.

[BGNC09] B. Bouvier, R. Grunberg, M. Nilges, and F. Cazals. Shelling the voronoi interface of protein-protein complexes reveals patterns of residue conservation, dynamics and composition. *Proteins: structure, function, and bioinformatics*, 76(3):677–692, 2009.

[Bil05] P. Bille. A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1-3):217–239, 2005.

[BK73] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, 1973.

[BW87] A.T. Brint and P. Willett. Algorithms for the identification of three-dimensional maximal common substructures. *Journal of Chemical Information and Computer Sciences*, 27(4):152–158, 1987.

[Caz10] F. Cazals. Revisiting the Voronoi description of protein-protein interfaces: Algorithms. In T. Dijkstra, E. Tsivtsivadze, E. Marchiori, and T. Heskes, editors, *IPAR International Conference on*

Pattern Recognition in Bioinformatics, pages 419–430, Nijmegen, the Netherlands, 2010. Lecture Notes in Bioinformatics 6282.

[CCJ99] L. Lo Conte, C. Chothia, and J. Janin. The atomic structure of protein-protein recognition sites. *Journal of Molecular Biology*, 285(5):2177 – 2198, 1999.

[CCLT09] P.M.M. De Castro, F. Cazals, S. Loriot, and M. Teillaud. Design of the cgal spherical kernel and application to arrangements of circles on a sphere. *Computational Geometry: Theory and Applications*, 42(6-7):536–550, 2009. Preliminary version as INRIA Tech report 6298.

[CJ02] P. Chakrabarti and J. Janin. Dissecting protein-protein recognition sites. *Proteins*, 47(3):334–43, 2002.

[CK08] F. Cazals and C. Karande. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407(1–3):564–568, 2008. INRIA Tech report 5615.

[CMJW03] R. Chen, J. Mintseris, J. Janin, and Z. Weng. A protein-protein docking benchmark. *Proteins*, 52:88–91, 2003.

[CPBJ06] F. Cazals, F. Proust, R. Bahadur, and J. Janin. Revisiting the voronoi description of protein-protein interfaces. *Protein Science*, 15(9):2082–2092, 2006.

[dBvKOS97] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1997.

[DMRW07] E. Demaine, S. Mozes, B. Rossman, and O. Weimann. An optimal decomposition algorithm for tree edit distance. *Automata, languages and programming*, pages 146–157, 2007.

[Fer99] A. Fersht. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. Freeman, 1999.

[FWE+11] S.J. Fleishman1, T.A. Whitehead1, D.C. Ekiert, C. Dreyfus, J.E. Corn, E-M. Strauch, I.A. Wilson, and D. Baker. Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science*, 332:816—821, 2011.

[GC05] M. Guharoy and P. Chakrabarti. Conservation and relative importance of residues across protein-protein interfaces. *Proc Natl Acad Sci U S A*, 102(43):15447–15452, 2005.

[HMWN02] I. Halperin, B. Ma, H. Wolfson, and R. Nussinov. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins*, 47(4):409–443, 2002.

[JBC08] J. Janin, R. P. Bahadur, and P. Chakrabarti. Protein-protein interaction and quaternary structure. *Quarterly reviews of biophysics*, 41(2):133–180, 2008.

[JT96] S. Jones and JM Thornton. Principles of protein-protein interactions. *PNAS*, 93(1):13–20, 1996.

[JT97] S. Jones and J.M. Thornton. Analysis of protein-protein interaction sites using surface patches1. *JMB*, 272(1):121–132, 1997.

[Kar72] R.M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations.*, 6:85–103, 06 1972.

[KJ10] J. Konc and D. Janezic. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics*, 26(9):1160, 2010.

[KN07] O. Keskin and R. Nussinov. Similar binding sites and different partners: Implications to shared proteins in cellular pathways. *Structure*, 15(3):341–354, 2007.

[KTWN04] O. Keskin, C.J. Tsai, H. Wolfson, and R. Nussinov. A new, structurally nonredundant, diverse data set of protein–protein interfaces and its implications. *Protein Science*, 13(4):1043–1055, 2004.

[LC10] S. Loriot and F. Cazals. Modeling macro-molecular interfaces with `intervor`. *Bioinformatics*, 26(7):964–965, 2010.

[LPLCT06] E.D. Levy, J.B. Pereira-Leal, C. Chothia, and S.A. Teichmann. 3D complex: a structural classification of protein complexes. *PLoS Comput Biol*, 2(11):e155, 2006.

[LW10a] M.F. Lensink and S.J. Wodak. Docking and scoring protein interactions: Capri 2009. *Proteins: Structure, Function, and Bioinformatics*, 78:3073–3084, 2010.

[LW10b] O. Lichtarge and A. Wilkins. Evolution: a guide to perturb protein function and networks. *Current opinion in structural biology*, 20(3):351–359, 2010.

[MDAY10] N. Malod-Dognin, R. Andonov, and N. Yanev. Maximum clique in protein structure comparison. In *International Symposium on Experimental Algorithms*, pages 106–117, 2010.

[ML07] I. Mihalek and O. Lichtarge. On itinerant water molecules and detectability of protein-protein int erfaces through comparative analysis of homologues. *J Mol Biol*, 369(2), 2007.

[Ö02] P.R.J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics.*, 120(1-3):197–207, 2002.

[RL02] M. Ruiz and M-P Lefranc. Imgt gene identification and colliers de perles of human immunoglobulins with known 3d structures. *Immunogenetics*, 53:857–883, 2002. 10.1007/s00251-001-0408-6.

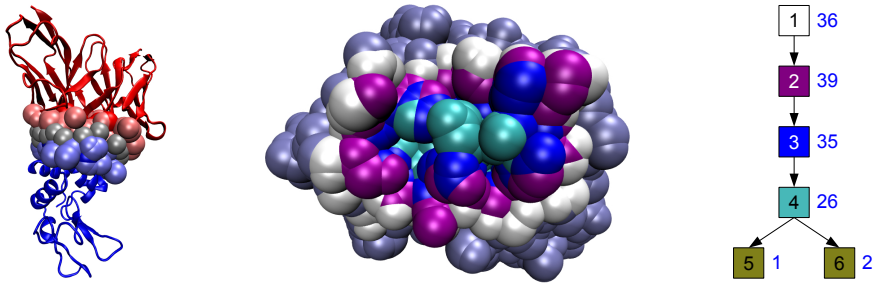
[RRA+05] D. Reichmann, O. Rahat, S. Albeck, R. Meged, O. Dym, and G. Schreiber. From The Cover: The modular architecture of protein-protein binding interfaces. *Proc Nat Acad Sci USA*, 102(1):57–62, 2005.

[Tar83] R. E. Tarjan. *Data Structures and Network Algorithms*, volume 44 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[WHKS06] C. Winter, A. Henschel, W.K. Kim, and M. Schroeder. SCOPPI: a structural classification of protein-protein interfaces. *Nucleic Acids Research*, 34(Database Issue):D310, 2006.

6 Artwork

Figure 3 Shelling a patch: illustration. (a) Side view of the protein complex 1vfb, with interface atoms displayed in red for partner A (chain A and B) and in blue for partner B (chain C). Grey atoms correspond to the interfacial water molecules. **(b,c)** Rotated view of the patch of partner *A*, and corresponding atom shelling tree. The colors of the atoms match those of the nodes of the shelling tree, the non interface atoms being represented in blue.



(a) (b) (c)

Figure 4 Shelling a patch: 2D illustration of the method. **(a)** The endpoint of the dashed purple edges, which are dual of the solid purple edges of the Voronoi diagram, identify the interface atoms. **(b)** The patch of the blue subunit is the Solvent Accessible Surface of its interface atoms, represented as solid circle arcs. Note that the packing defect in-between the atoms centered at b_1, b_2, b_4, b_5 is such that the patch has two connected components cc_1 and cc_2. **(c)** A packing defect in-between the partners dismisses atom a_3 as interface atom. **(d)** On this 2D example, the patch has two c.c.; when the same occurs in 3D, the patch is connected but is topologically equivalent to an annulus i.e. has two rims called *connected component of the boundary* or CCB. The largest one only is used to initiate the shelling order calculation.

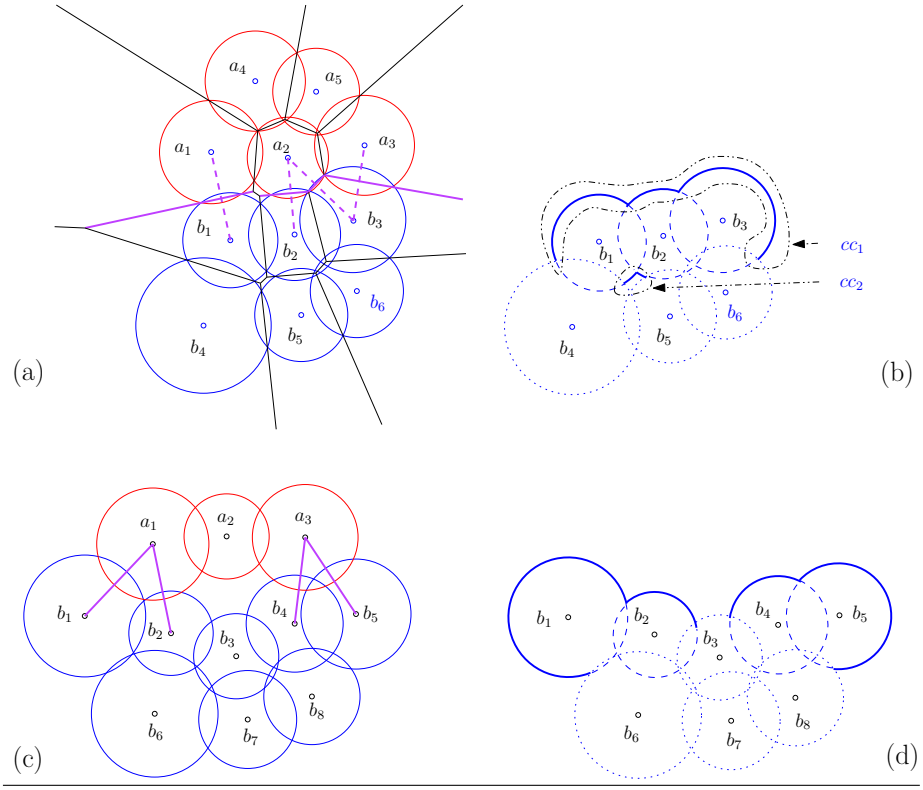


Figure 5 Morphology of the patches: number of atoms versus number of shells. The crosses identify the canonical morphologies of presented on Fig. 6.

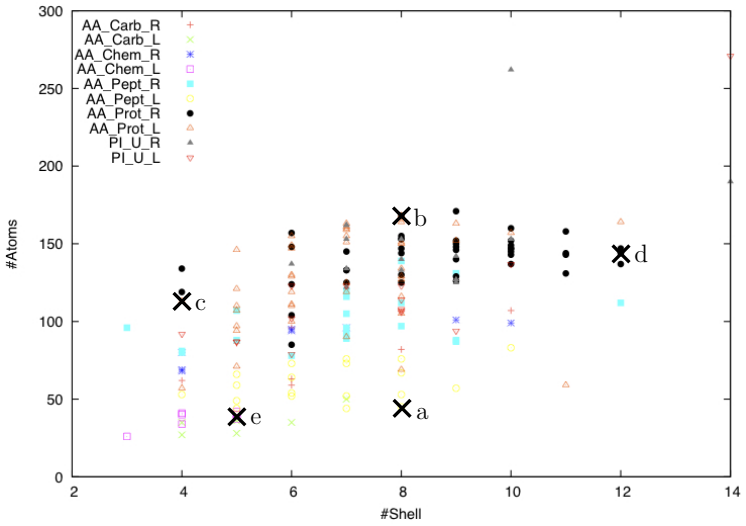


Figure 6 Illustration of five canonical morphologies. The ligands are represented as cartoons when they do not clutter the picture. **(a)** tubular (pdbid 3eys, chain B); **(b)** pyramidal (pdbid 3a6c, chain B). **(c)** anisotropic (pdbid 3h0t, chain A) **(d)** isotropic (pdbid 2ih3, chain A); **(e)** pear-like (pdbid 2dqu, chain B)

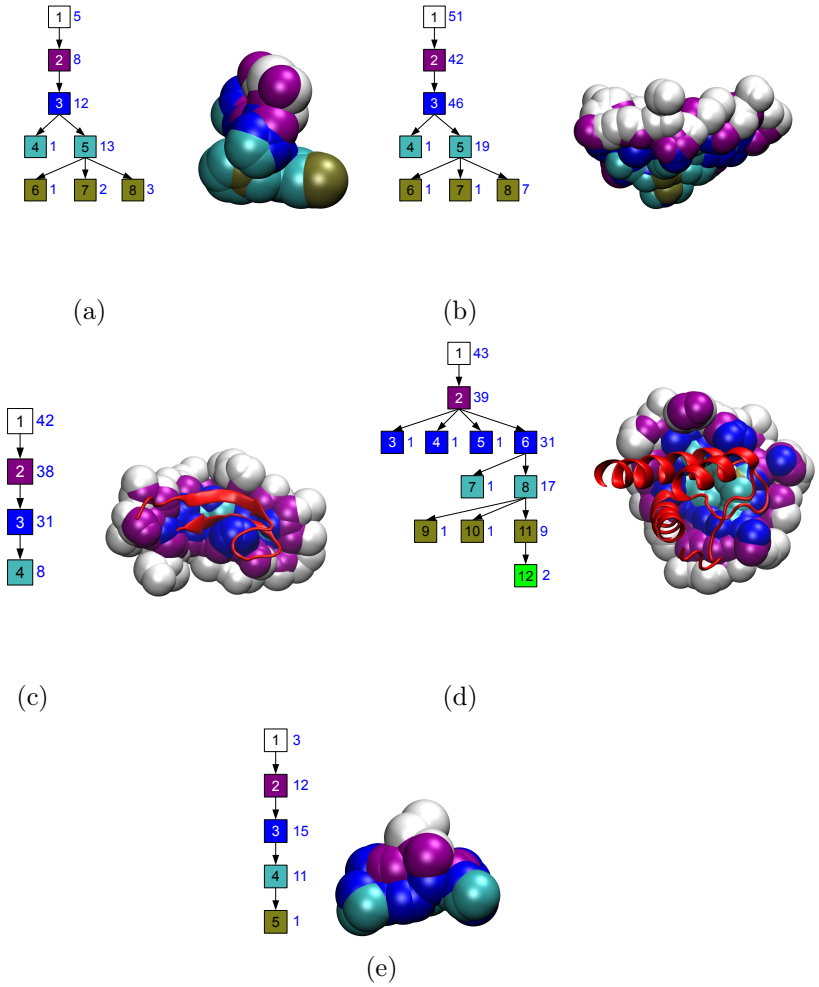


Figure 7 Signature of the five morphologies of Fig. 6. Each morphology is characterized by a specific variation of the number of atoms at each SO of the atom shelling tree. Pear-like binding patch (in red): very few atoms at SO zero, then, as the SO increases, the number of atoms significantly increases, and then decreases. Tubular patch (green): small number of atoms at each SO, without large variation. For the pyramidal, isotropic and anisotropic patches, the number of atoms is very large at SO zero, and then the number of atoms decreases as the SO increases. Isotropic patch (in pink) reaches large SO, while anisotropic binding patch (in light-blue) is less profound.

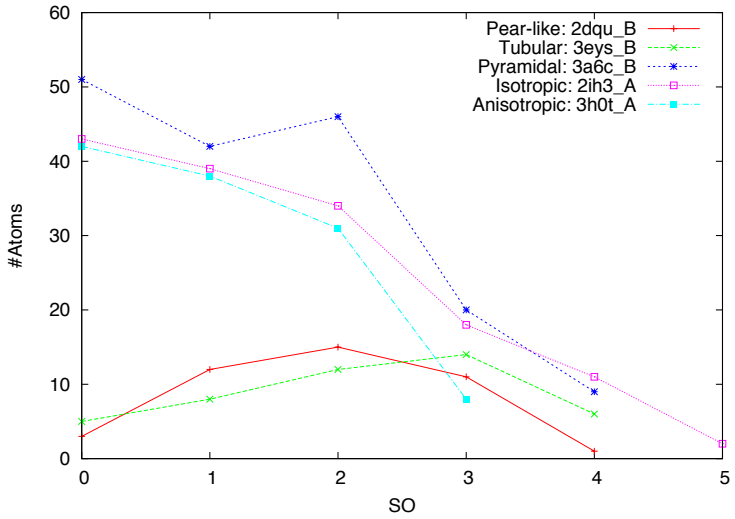


Figure 8 Topological (DIS_t) versus geometrical (DIS_g) dissimilarities. With one point per pairwise comparison, the plot illustrates the fact that the topological dissimilarity is an upper-bound of the geometrical dissimilarity. The black rectangle singles out instances having similar topologies (low DIS_t) but different geometries (high DIS_g).

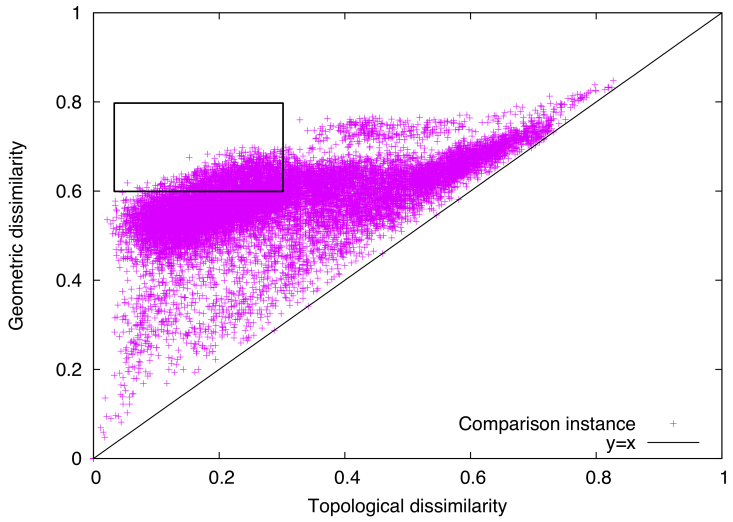


Table 1 Identification rates via the typed bio-family of the most similar patch. For each method, columns 2 to 4 present, in percentage, the number times $\hat{p}$ comes from the family of p (column 2), from the partner family of p (column 3), or from an unrelated family (column 4).

Method	case I: $\hat{p} \in P - \{p\}$	case II: $\hat{p} \in \overline{P}$	case III: $\hat{p} \in P^c$
TED_t	47.2%	11.4%	41.4%
TED_g: $\epsilon = 1$Å	69.0%	4.9%	26.1%
TED_g: $\epsilon = 1.5$Å	68.5%	2.2%	29.3%
TED_g: $\epsilon = 2$Å	73.4%	3.8%	22.8%

Figure 9 Identifications having low geometric dissimilarities are correct. The number of identified patches are plotted as a function of the geometric dissimilarity between the queries and their nearest neighbors $(p, \hat{p})$. When this dissimilarity is below 0.24 (i.e. more than 76% of isometric atoms), the identification is correct. Then, the higher is the associated dissimilarity, the higher is the error rate.

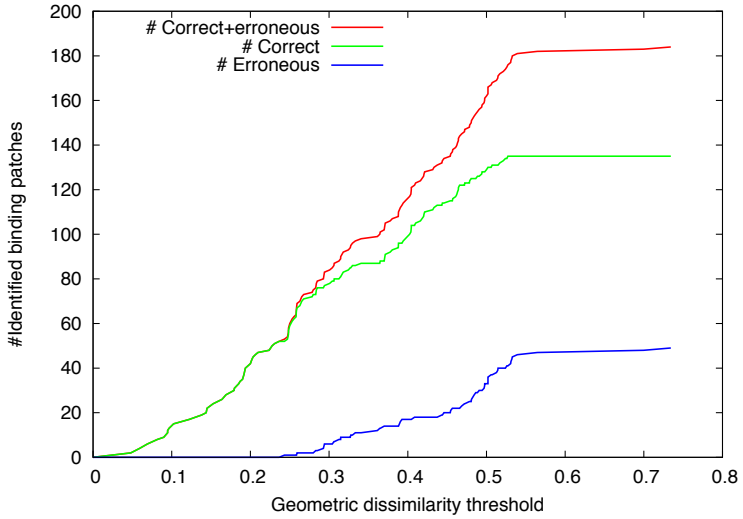
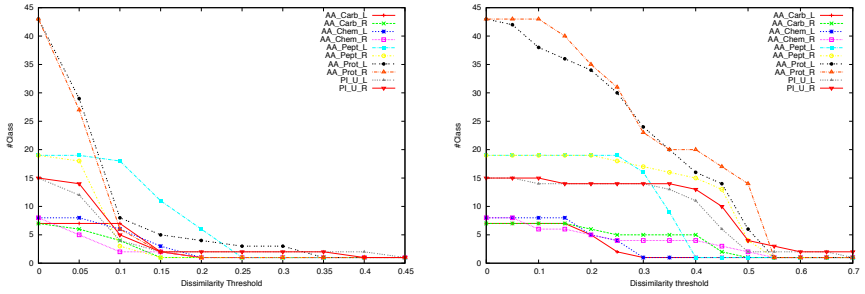


Figure 10 Clustering of the typed bio-families at different dissimilarity threshold. Left: Number of sub-classes found by the topological clustering when the dissimilarity threshold δ_t varies in [0, .45]. Right: Number of sub-classes found by the geometrical clustering when the dissimilarity threshold δ_g varies in [0, .7].



7 Supplement

7.1 Dataset of Biological Complexes

The supplemental Table 1 displays the PDB ids and the partner specifications of our dataset of 92 high-resolution complexes (resolution < 2), while the supplemental Table 2 overviews the typed bio-families of the corresponding database of 184 patches.

PDB Id	Chains		Typed bio-families	
	Partner A	Partner B	Partner A	Partner B
1Q9Q	BA	C	AA_Carb_R	AA_Carb_L
1Q9R	BA	C	AA_Carb_R	AA_Carb_L
1S3K	HL	C	AA_Carb_R	AA_Carb_L
1ZLS	HL	X	AA_Carb_R	AA_Carb_L
3HNS	HL	1	AA_Carb_R	AA_Carb_L
3HNT	HL	1	AA_Carb_R	AA_Carb_L
3HNV	HL	1	AA_Carb_R	AA_Carb_L
1HYX	HL	1	AA_Chem_R	AA_Chem_L
1HYY	HL	1	AA_Chem_R	AA_Chem_L
1N7M	LH	1	AA_Chem_R	AA_Chem_L
2DQT	HL	1	AA_Chem_R	AA_Chem_L
2DQU	HL	1	AA_Chem_R	AA_Chem_L
2R23	BA	1	AA_Chem_R	AA_Chem_L
2R2B	BA	1	AA_Chem_R	AA_Chem_L
3HZV	BA	1	AA_Chem_R	AA_Chem_L
1CE1	HL	P	AA_Pept_R	AA_Pept_L
1E4W	HL	P	AA_Pept_R	AA_Pept_L
1I8K	BA	C	AA_Pept_R	AA_Pept_L
1MVU	BA	P	AA_Pept_R	AA_Pept_L
1TJG	HL	P	AA_Pept_R	AA_Pept_L
1U8I	BA	C	AA_Pept_R	AA_Pept_L
2B1H	HL	P	AA_Pept_R	AA_Pept_L
2F5B	HL	P	AA_Pept_R	AA_Pept_L
2FX7	HL	P	AA_Pept_R	AA_Pept_L
3DRQ	BA	C	AA_Pept_R	AA_Pept_L
3EYS	HL	Q	AA_Pept_R	AA_Pept_L
3FN0	HL	P	AA_Pept_R	AA_Pept_L
3G5Y	BA	E	AA_Pept_R	AA_Pept_L
3GO1	HL	P	AA_Pept_R	AA_Pept_L
3IDG	BA	C	AA_Pept_R	AA_Pept_L
3IFL	HL	P	AA_Pept_R	AA_Pept_L
3LEY	HL	P	AA_Pept_R	AA_Pept_L
3MLR	HL	P	AA_Pept_R	AA_Pept_L
3MNZ	BA	P	AA_Pept_R	AA_Pept_L
1A2Y	BA	C	AA_Prot_R	AA_Prot_L
1DQJ	BA	C	AA_Prot_R	AA_Prot_L
1F58	HL	P	AA_Prot_R	AA_Prot_L
1FNS	HL	A	AA_Prot_R	AA_Prot_L
1G7H	BA	C	AA_Prot_R	AA_Prot_L
1G7I	BA	C	AA_Prot_R	AA_Prot_L
1G7J	BA	C	AA_Prot_R	AA_Prot_L
1G7L	BA	C	AA_Prot_R	AA_Prot_L
1G7M	BA	C	AA_Prot_R	AA_Prot_L

PDB Id	Chains		Typed bio-families	
	Partner A	Partner B	Partner A	Partner B
1IQD	BA	C	AA_Prot_R	AA_Prot_L
1J1O	HL	Y	AA_Prot_R	AA_Prot_L
1J1P	HL	Y	AA_Prot_R	AA_Prot_L
1J1X	HL	Y	AA_Prot_R	AA_Prot_L
1JPS	HL	T	AA_Prot_R	AA_Prot_L
1K4C	AB	C	AA_Prot_R	AA_Prot_L
1KIQ	BA	C	AA_Prot_R	AA_Prot_L
1KIR	BA	C	AA_Prot_R	AA_Prot_L
1NBY	BA	C	AA_Prot_R	AA_Prot_L
1NBZ	BA	C	AA_Prot_R	AA_Prot_L
1NDG	BA	C	AA_Prot_R	AA_Prot_L
1ORS	BA	C	AA_Prot_R	AA_Prot_L
1OSP	HL	O	AA_Prot_R	AA_Prot_L
1R3J	BA	C	AA_Prot_R	AA_Prot_L
1UA6	HL	Y	AA_Prot_R	AA_Prot_L
1UAC	HL	Y	AA_Prot_R	AA_Prot_L
1WEJ	HL	F	AA_Prot_R	AA_Prot_L
1YQV	HL	Y	AA_Prot_R	AA_Prot_L
2ADF	HL	A	AA_Prot_R	AA_Prot_L
2DQC	HL	Y	AA_Prot_R	AA_Prot_L
2DQD	HL	Y	AA_Prot_R	AA_Prot_L
2DQE	HL	Y	AA_Prot_R	AA_Prot_L
2DQI	HL	Y	AA_Prot_R	AA_Prot_L
2DQJ	HL	Y	AA_Prot_R	AA_Prot_L
2IH3	AB	C	AA_Prot_R	AA_Prot_L
2VXQ	HL	A	AA_Prot_R	AA_Prot_L
2VXT	HL	I	AA_Prot_R	AA_Prot_L
3A67	HL	Y	AA_Prot_R	AA_Prot_L
3A6B	HL	Y	AA_Prot_R	AA_Prot_L
3A6C	HL	Y	AA_Prot_R	AA_Prot_L
3BAE	HL	A	AA_Prot_R	AA_Prot_L
3D9A	HL	C	AA_Prot_R	AA_Prot_L
3FFD	AB	P	AA_Prot_R	AA_Prot_L
3H0T	BA	C	AA_Prot_R	AA_Prot_L
1acb	I	E	PI_U_L	PI_U_R
1avw	B	A	PI_U_L	PI_U_R
1cse	I	E	PI_U_L	PI_U_R
1ppe	I	E	PI_U_L	PI_U_R
1spb	P	S	PI_U_L	PI_U_R
1tgs	I	Z	PI_U_L	PI_U_R
2ptc	I	E	PI_U_L	PI_U_R
2sic	I	E	PI_U_L	PI_U_R
2tec	I	E	PI_U_L	PI_U_R
1dan	LH	TU	PI_U_R	PI_U_L
1fle	E	I	PI_U_R	PI_U_L
1mct	A	I	PI_U_R	PI_U_L
1ppf	E	I	PI_U_R	PI_U_L
3sgb	E	I	PI_U_R	PI_U_L
3tpi	Z	I	PI_U_R	PI_U_L

Supplemental Table 1: **The 92 protein complexes used in this study.**

Supplemental Table 2 The typed bio-family classification of the 184 patches from our database.

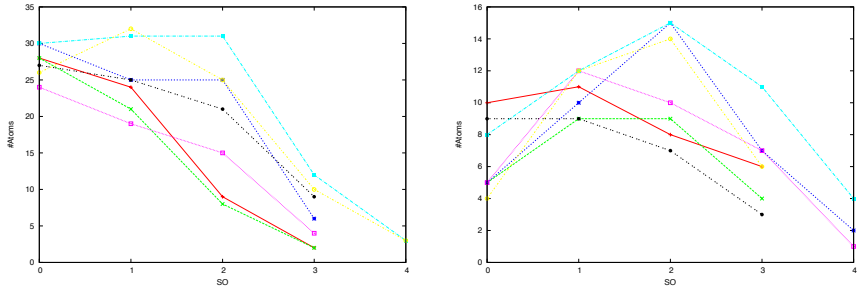
Family of complex	Bio-Family of complex	Partner type	Class identifier	#patches
(AA) Antibody / Antigen	(Carb) Carbohydrate	(R) Receptor	AA_Carb_R	7
		(L) Ligand	AA_Carb_L	7
	(Chem) Chemical	(R) Receptor	AA_Chem_R	8
		(L) Ligand	AA_Chem_L	8
	(Pept) Peptide	(R) Receptor	AA_Pept_R	19
		(L) Ligand	AA_Pept_L	19
	(Prot) Protein	(R) Receptor	AA_Prot_R	43
		(L) Ligand	AA_Prot_L	43
(PI) Protease / Inhibitor	(U) Unknown	(L) Ligand	PI_U_L	15
		(R) Receptor	PI_U_R	15

7.2 Morphological Analysis: Topology versus Geometry

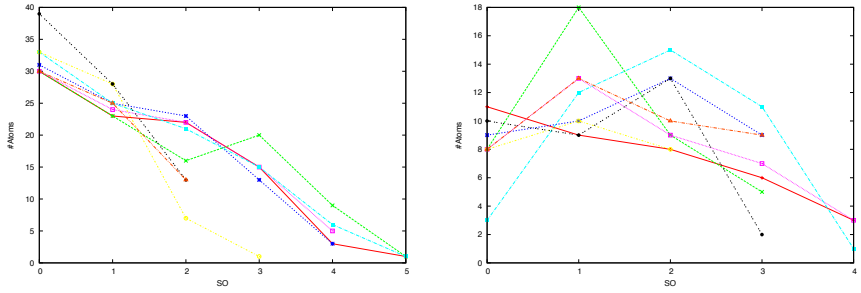
7.2.1 Size of shells as a function of the Shelling Order

In section 3.1, we discussed the variation of the size of shells as a function of the SO, for selected morphologies. The supplemental Figures 1, 2, 3, 4 and 5 provide all such profiles for the patches of our database.

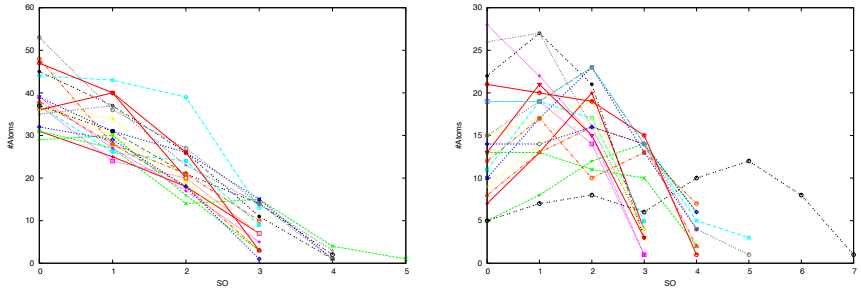
Supplemental Figure 1 For each patch of the considered typed bio-family, the number of atom at a given SO value is plotted against the SO value. **Left:** the AA_Carb_R family, **Right:** the AA_Carb_L family,



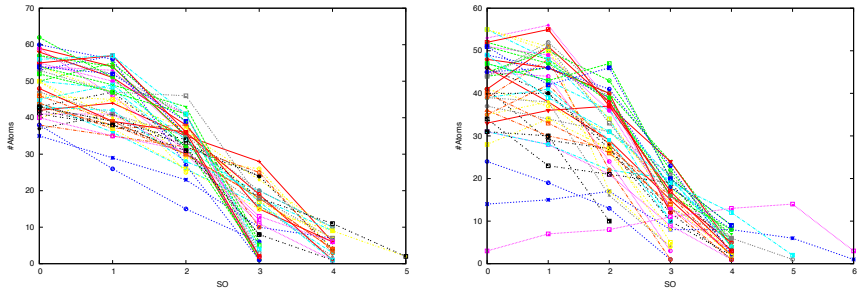
Supplemental Figure 2 For each patch of the considered typed bio-family, the number of atom at a given SO value is plotted against the SO value. **Left:** the AA_Chem_R family, **Right:** the AA_Chem_L family.



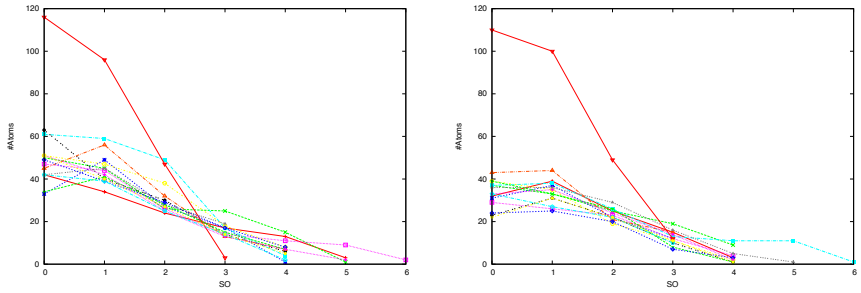
Supplemental Figure 3 For each patch of the considered typed bio-family, the number of atom at a given SO value is plotted against the SO value. **Left:** the AA_Pept_R family, **Right:** the AA_Pept_L family.



Supplemental Figure 4 For each patch of the considered typed bio-family, the number of atom at a given SO value is plotted against the SO value. **Left:** the AA_Prot_R family, **Right:** the AA_Prot_L family.



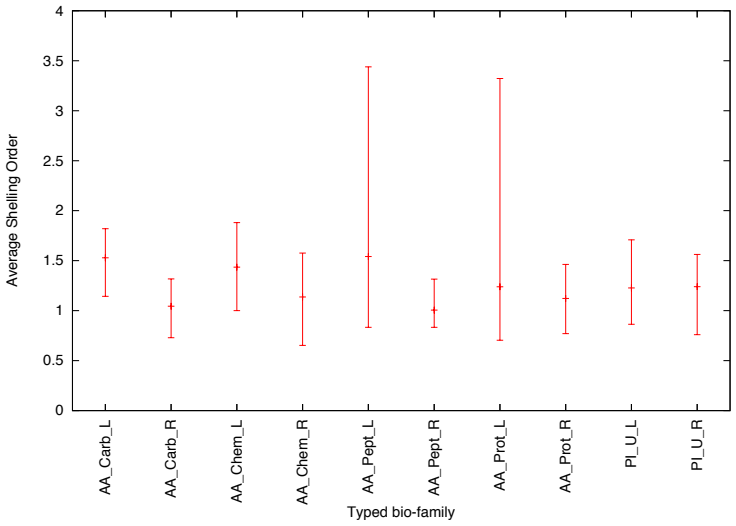
Supplemental Figure 5 For each patch of the considered typed bio-family, the number of atom at a given SO value is plotted against the SO value. **Left:** the PI_U_R family, **Right:** the PI_U_L family.



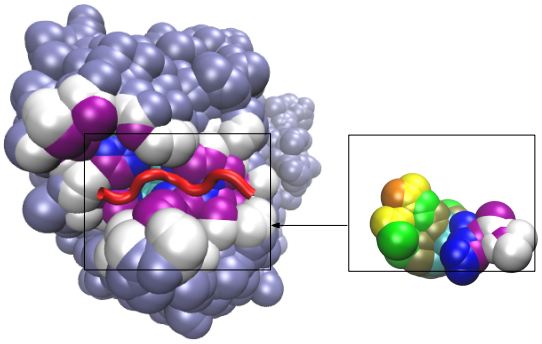
7.2.2 Average Shelling Order

In section 3.1, we mentioned that the average shelling order is a fingerprint of the asymmetry of two patches in a complex. The supplemental Figure 7 illustrates this property.

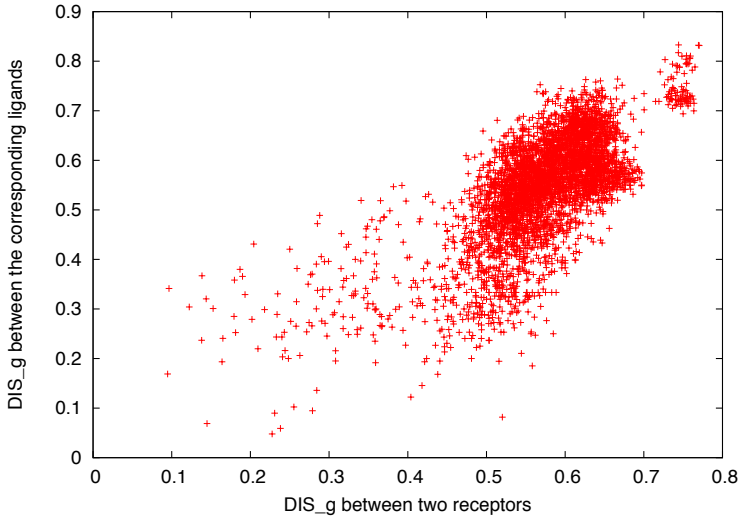
Supplemental Figure 6 The average shelling order ($\overline{SO}$) assesses that most partner patches are asymmetric. For each typed bio-family, the minimum, average and maximum $\overline{SO}$ are plotted next to the $\overline{SO}$ of their partner families. Between any two partners typed bio-families there is a large discrepancy in the $\overline{SO}$ values.



Supplemental Figure 7 An example of complex whose asymmetry is revealed by the average Shelling Order. Complex 3ifl, from AA_Pept. Left: the receptor (3ifl partner A). Non interface atoms are displayed in grey, and partner B is displayed with a cartoon representation. Right: the ligand (3ifl partner B) alone. The SO of the partners receptor and ligand respectively vary in the range $0 \ldots 4$ and $0 \ldots 7$. The receptor has an $\overline{SO}$ of 1.17, while the ligand has an $\overline{SO}$ of 3.47.

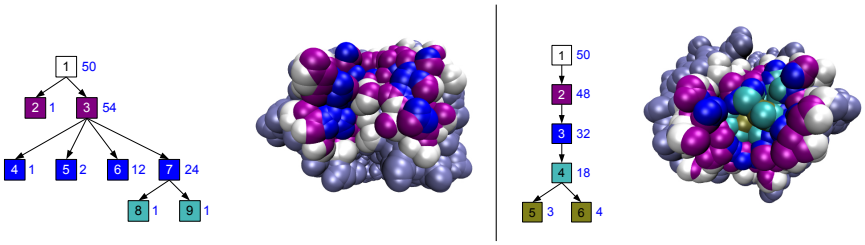


Supplemental Figure 8 Similar receptors have similar ligands. For any two receptors, their geometric dissimilarity is plotted against the geometric dissimilarity between the two corresponding ligands. The dissimilarity between two receptors correlates with the dissimilarity between the corresponding ligands.



7.2.3 Geometry versus Topology

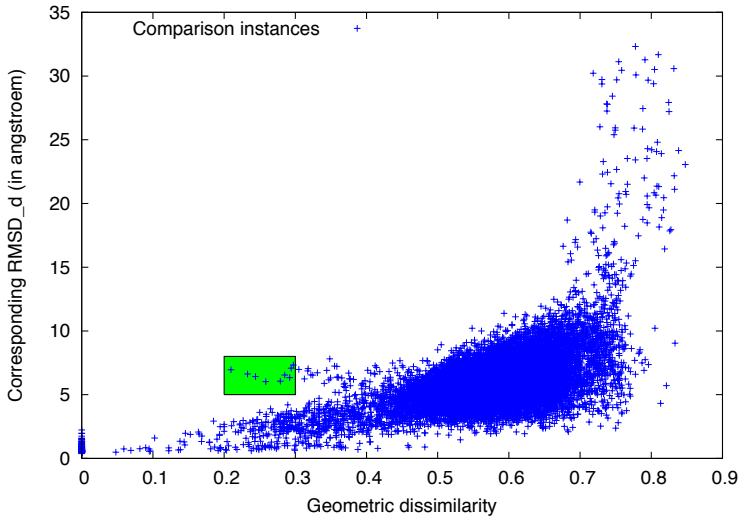
Supplemental Figure 9 Two patches having similar topologies but different geometries. Left: patch of 1dqj partner A. Right: patch of 1jps partner A. Their topological dissimilarity DIS_t is about 0.18 (due to shells 1-3-7 from 1dqj_A that match shells 1-2-3 from 1jps_A), but their geometrical dissimilarity DIS_g is about 0.68.



7.2.4 Geometrical dissimilarity and $RMSD_d$

Algorithm TED_g provides quasi-isometric matchings at the shell level, but does not provide any guarantee on the $RMSD_d$ at the patch level, which calls for the investigation of the correlation between DIS_g ($\epsilon = 2$Å) and the $RMSD_d$ of the matching. The corresponding plot for the $\binom{184}{2} + 184 = 17020$ comparisons is presented on the supplemental Figure 10. The $RMSD_d$ globally increases with the geometric dissimilarity, and so does the variance of the $RMSD_d$ for a fixed DIS_g. This plot also exhibits two subsets of instances of particular interests: those characterized by $RMSD_d \geq 15$Å), and those with a low dissimilarity scores but large $RMSD_d$ values. The instances of the first subset all involve one of the two patches of the complex 1dan, which are different from all the others patches. All such instances are characterized by $\text{DIS}_g \geq 0.69$, corresponding to at most 31% of atoms in common This nine instances of the second set are characterized by $\text{DIS}_g \leq 0.3$ and $RMSD_d \geq 5$. All these instances involve the following eight patches : 1j1o_A, 1j1p_A, 1j1x_A, 2dqc_A, 2dqd_A,2dqe_A, 3a67_A and 3d9a_A, that all come from the AA_Prot_R typed bio-family and that all bind Lysozyme C.

Supplemental Figure 10 Geometric dissimilarity (DIS_g) versus RMSD of internal distances ($RMSD_d$). The green region singles out instances with low geometric dissimilarity but high $RMSD_d$ values.



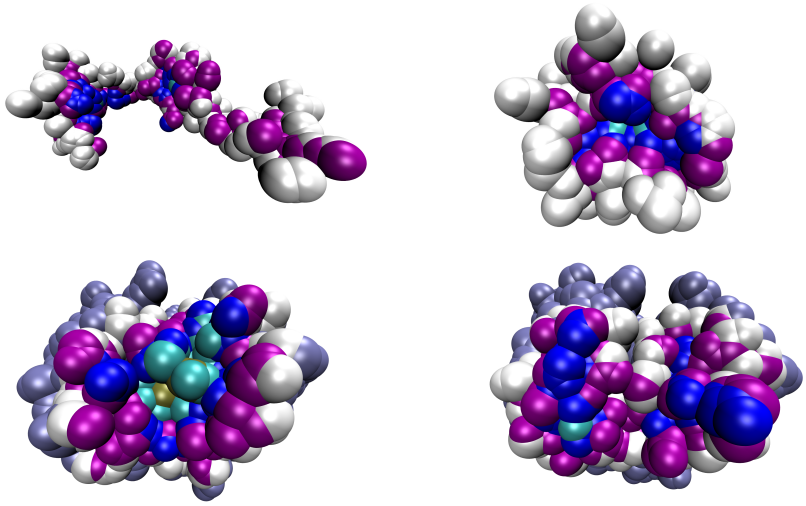
7.3 Geometric and topological descriptors versus biological functions

7.3.1 Family identification

Supplemental Table 3 Percentage of correct identification per typed bio-family. The typed bio-family of a patch is identified by the typed bio-family of its nearest neighbors $\hat{p}$, according to the geometric dissimilarity (with a distance threshold of 2Å). Columns 2 (resp. 3) presents for each typed bio-family the number (resp percentage) of correctly identified binding patches.

Typed bio-family	Correctly identified patches	percentage
AA_Carb_R	6/7	85.7%
AA_Carb_L	6/7	85.7%
AA_Chem_R	6/8	75.0%
AA_Chem_L	6/8	75.0%
AA_Pept_R	11/19	57.9%
AA_Pept_L	7/19	36.8%
AA_Prot_R	35/43	81.4%
AA_Prot_L	35/43	81.4%
PI_U_R	12/15	80.0%
PI_U_L	11/15	73.3%

Supplemental Figure 11 Two examples of the inconsistencies in the original typed bio-family classification. Top: both the 1dan receptor (partner A) and the 1mct receptor (partner A) are classified as PI_U_R patches but their 3D structure is visibly different. Bottom: both the 1jps receptor (partner A) and the 1j1p receptor (partner A) are classified as AA_Prot_R patches. Their structures are also different.



Robustness to noise. To study the noise effect on the identifications of TED_g, we compared the number of correct identification obtained on two databases containing the same four typed bio-families, namely "AA_Pept_R", "AA_Pept_L", "AA_Prot_R" and "AA_Prot_L", represented in each database by the same number of patches, but one is composed of high resolution ($\leq$ 2Å) patches, and the other one is composed of low resolution (between 2Å and 3Å). The supplemental Table 4 shows that the number of correct identifications is about 77.4% when using high resolution patches versus 70.1% when using low resolution patches. Note that in both cases, all identification associated to small dissimilarity scores were correct (the first error appears with a dissimilarity score of 0.25).

Supplemental Table 4 Correct identifications at different resolutions. Columns 2 presents for each typed bio-family the number of correctly identified patches in the high resolution database, and column 3 presents the same for the low resolution database. The typed bio-family of a patch is identified by the typed bio-family of its nearest neighbors $\hat{p}$, according to the dissimilarity score of TED_g.

	Correctly identified patches	
Typed bio-family	High resolution	Low resolution
AA_Pept_R	14/19	18/19
AA_Pept_L	11/19	10/19
AA_Prot_R	36/43	29/43
AA_Prot_L	35/43	30/43

Supplemental Table 5 Probabilities that the lists of pairwise scores (obtained by TED_g with ϵ=2Å) come from the same distribution law. The very low probability scores show that the null hypotheses (same distribution laws) are false.

Family (=P)	(P, P) vs $(P, \overline{P})$	(P, P) vs (P, P^C)
AA_Carb_R	3.76e-06	3.02e-07
AA_Carb_L	5.15e-11	1.27e-13
AA_Chem_R	1.42e-08	1.30e-08
AA_Chem_L	3.44e-14	5.78e-17
AA_Pept_R	1.80e-17	1.31e-27
AA_Pept_L	9.47e-69	9.78e-70
AA_Prot_R	7.25e-04	3.93e-38
AA_Prot_L	2.86e-56	9.73e-49
PI_U_L	2.76e-23	6.25e-20
PI_U_R	7.10e-06	1.14e-14

7.3.2 Analysis of biological families

The evolution of the number of unclassified elements (i.e. that are alone in their class), that is presented in the supplemental Figure 12) leads to the same conclusion.

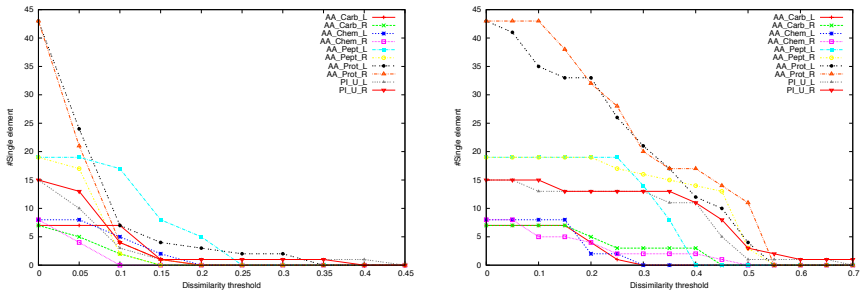
Geometric clustering and biological function. The correctness of the identifications performed with $\epsilon_g \leq 0.24$ suggests using this threshold to cluster the correctly identified patches. (At a threshold of $\delta_g = 0.24$, there are 54 such patches.)

The supplemental Table 6 presents the correlation between the corresponding automatic classification and the original typed bio-family classification.

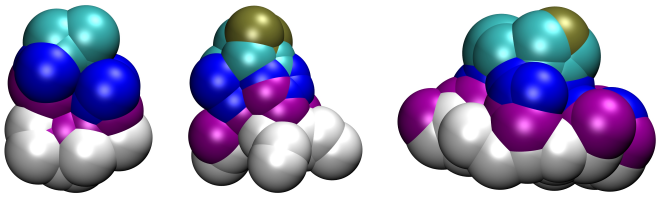
Parameter ϵ_g affects the automatic classification in the following ways: the smaller is ϵ_g, the larger is the number of classes that we obtain (including the number of unclassified patches that are alone in their classes).

Small values of ϵ_g also increase the number of typed bio-families that are split into many classes (like in supplemental Table 6, where the AA_Prot_R typed bio-family is divided into classes I, J and K). On the opposite, large values of ϵ_g relate to small number of classes, implying that many typed bio-families are merged into a single class (like in the supplemental Table 6, where the AA_Carb_L, the AA_Chem_L and the AA_Pept_L typed bio-families are merged into class C)

Supplemental Figure 12 Number of single/unclassified elements at different dissimilarity threshold. Left: Number of unclassified elements by the topological clustering at different dissimilarity threshold δ_t values. Right: Number of unclassified elements by the geometrical clustering at different dissimilarity threshold δ_g values.



Supplemental Figure 13 Example of the small ligand patches that are putted into the same class. Left: a AA_Carb_L patch (1q9q_B). Middle: a AA_Chem_L patch (1hyx_B). Right: a AA_Pept_L patch (1e4w_B).



Supplemental Table 6 Automatic versus typed bio-family classification. Each row in the table represent a given class from the typed bio-family classification, and each column represent one of the classes of the automatic classification that is obtained by using a threshold $\epsilon = 0.24$ on the geometric dissimilarity. Note that the automatic classification also contains 130 unclassified elements that are not represented here. Each entry in the table count the number of patches belonging to the corresponding typed bio-family and automatic classes. The automatic classification over separates receptor binding patches into many classes (ex: the AA_Prot_R in classes I, J and K), but clusters many small ligands into the same class (in class C). Example of such ligand are given in the supplemental Figure 13.

Typed bio-family														
AA_Carb_L			5											
AA_Carb_R	2	2												
AA_Chem_L			4	2										
AA_Chem_R					4									
AA_Pept_L			1											
AA_Pept_R														
AA_Prot_L						2	10	3						
AA_Prot_R									5	2	8			
PI_U_L												2		
PI_U_R													2	
	A	B	C	D	E	F	G	H	I	J	K	L	M	Classes

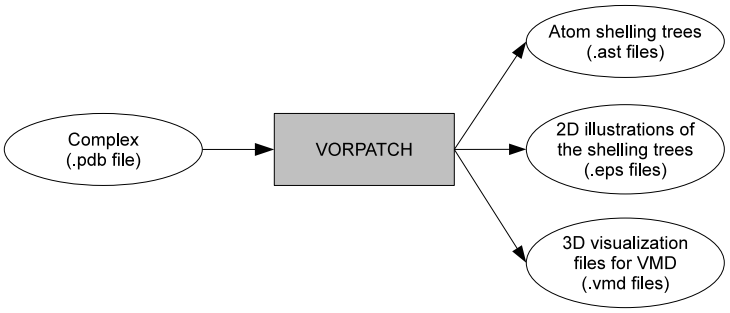
7.4 Software: application and file formats

In this section, we describe the tools that we designed for generating and comparing atom shelling trees. The two softwares, VORPATCH and COMPATCH, are available from `http://cgal.inria.fr/abs/vorpatch-compatch/`.

7.4.1 Program VORPATCH

Given the 3D structure of a complex (a .pdb file), and the two sets of chain IDs of the considered partners, VORPATCH generates the atom shelling trees of the two patches using the algorithm sketched in section 5.1.3. The atom shelling trees are recorded in the custom file format described in the supplemental Fig. 15. VORPATCH also generates encapsulated postscript figures (.eps file format) of the atom shelling trees, as well as 3D visualization files (.vmd) for VMD `http://www.ks.uiuc.edu/Research/vmd/`. (To load these vmd files, the *fastload* plugin available from the aforementioned web site is highly recommended.)

Supplemental Figure 14 Overview of the patch generation with COMPATCH**.**



Supplemental Figure 15 The atom shelling tree (.ast) file format. An atom shelling tree is described by the list of its nodes (or shells). Each node is first described by a header line containing three integers: the node ID (starting from 1), the number of atom of the corresponding shell, and the node's father ID (0 if a node is a root one). A header line is then followed by lines describing the shell's atoms: one line per atoms, each containing the pdb ID of the atom, its x, y and z coordinates and its expanded radius.

```
# header of the first node (node ID, #atoms, father ID)
1 2 0
# pdb IDs (pid) coordinates (x, y, z) and radii (r) of the first node's atoms:
# pid x y z r
2165 68.109 72.871 103.635 3.27
1921 59.09 85.686 95.602 3.27
# header of the second node
2 3 1
# pdb IDs coordinates and radii of the second node's atoms
1966 73.249 81.239 101.172 3.27
1920 61.252 84.098 94.165 2.8
1927 63.162 85.856 93.171 3.27
# ...
```

7.4.2 Program COMPATCH

Supplemental Figure 16 Overview of the patch comparison with COMPATCH**.**

Given two patches (.ast file format), COMPATCH use the tree-edit-distance based methods presented in section 5.2.1 to measure their dissimilarity, and also record

the optimum tree-edit-script (the sequence of tree-edit operations) in the custom file format described in the supplemental Fig. 17. The numerical values (dissimilarity scores, size of the two trees, tree-edit-distance values and running times) are printed into the console or the log file.

Supplemental Figure 17 The tree-edit-script (.tes) file format. The first line recall the filename of the two input atom shelling trees. The consecutive lines present the optimum tree-edit script (sequence of tree-edit operations) for transforming the first tree into the second one, and for each operation the associated cost is given. If the comparison was done with TED_g, then the mapping operations are followed by the corresponding lists of atom matchings.

```
./test/1a3r_A.ast ./test/1a3r_B.ast
Delete node 1 from tree 1, cost = 3
Delete node 1 from tree 2, cost = 1
Map node 2 from tree 1 with node 2 from tree 2, cost = 4
    1932 (61.261 85.936 90.979)  <->  3454 (63.623 66.66 98.256)
    2098 (70.15 77.144 100.862)  <->  3393 (74.594 76.314 94.763)
    2100 (69.981 76.698 98.432)  <->  3394 (72.268 75.409 94.954)
Map node 3 from tree 1 with node 3 from tree 2, cost = 3
    1967 (72.941 80.456 99.934)  <->  3432 (69.053 75.438 91.536)
```

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399