Characterizing the Morphology of Protein Binding Patches

Noël Malod-Dognin, Achin Bansal, Frédéric Cazals

Characterizing the Morphology of Protein Binding Patches

Noël Malod-Dognin, Achin Bansal, Frédéric Cazals

RESEARCH REPORT
N° 7743
September 2011
Project-Team ABS

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

Characterizing the Morphology of Protein Binding Patches

Noël Malod-Dognin, Achin Bansal, Frédéric Cazals

Project-Team ABS

Research Report n° 7743 — version 2 — initial version September 2011 — revised version June 2012 — 49 pages

Abstract: Let the patch of a partner in a protein complex be the collection of atoms accounting for the interaction. To improve our understanding of the structure-function relationship, we present a patch model decoupling the topological and geometric properties. While the geometry is classically encoded by the atomic positions, 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 comparing patches at the shell level, which may favor topological or geometric features.
On the biological side, we address four questions, using 249 co-crystallized heterodimers organized in biological families. First, we dissect the morphology of binding patches, and show that Nature enjoyed the topological and geometric degrees of freedom independently while retaining a finite set of qualitatively distinct topological signatures. Second, we argue that our shell-based comparison is effective to perform atomic-level comparisons, and show that topological similarity is a less stringent than geometric similarity. We also use the topological versus geometric duality to exhibit topo-rigid patches, whose topology (but not geometry) remains stable upon docking. Third, we use our comparison algorithms to infer specificity related information amidst a database of complexes. Finally, we exhibit a descriptor outperforming its contenders to predict the binding affinities of the affinity benchmark.
The softwares developed with this paper are available from http://team.inria.fr/abs/vorpatch_compatch/.

Key-words: Protein complex, binding patch, interface morphology, structural comparisons; Voronoi models, shelling tree, tree edit distance, dynamic programming based comparisons.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Caractérisation de la morphologie des patchs de liaisons protéiques

Résumé : Définissons le patch d'un complexe protéique comme l'ensemble des atomes rendant compte de l'interaction. Afin d'éclairer le rapport structure - fonction, nous proposons un modèle de patch découplant les propriétés topologiques et géométriques. La géométrie étant comme à accoutumée codée par les positions des atomes, la topologie est encodée au moyen d'un graphe indiquant les positions relatives de couches concentriques partitionnant les atomes du patch. Ce codage fournit la base d' un algorithme générique de comparaison de patchs au niveau atomique, lequel peut être instantié pour privilégier une comparaison de la géométrie ou de la topologie.

Du point de vue biologique, nous examinons quatre questions à l'aide d'une base de données de 249 structures de complexes co-cristalisés, organisée en familles biologiques. Premièrement, nous montrons que la Nature utilise les degrés de liberté topologiques et géométriques indépendamment, tout en ne conservant qu'un ensemble fini de signatures topologiques qualitativement distinctes. Deuxièmement, nous montrons l'efficacité de nos méthodes à produire des comparaisons au niveau atomique, et nous observons que la similarité topologique est une notion moins stricte que ne l'est la similarité géométrique habituellement utilisée. Nous utilisons également la dualité entre la topologie et la géométrie pour caractériser des patchs topo-rigides, dont la topologie (mais pas la géométrie) reste stable lors de l'amarrage. Troisièmement, nous utilisons nos algorithmes pour étudier la spécificité d'interactions observées au sein de notre base de données. Enfin, nous proposons un descripteur améliorant la prédiction des constantes d'affinité des complexes de *l'affinity benchmark*.

Les logiciels VORPATCH et COMPATCH développés pour réaliser ce travail sont disponibles via http://team.inria.fr/abs/vorpatch_compatch/.

Mots-clés : Complexes protéiques, patchs de liaisons, morphologie des interfaces, comparaisons de structures, modèles de Voronoi, arbre de couches, distance d'édition d'arbre, programmation dynamique.

Contents

1 Introduction

1.1 Modeling Protein Binding Patches, with Applications

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 [1], but also pave the way to protein engineering [2]. Structural information is also complemented by directed mutagenesis and binding affinity measurements, which convey information of biological and thermodynamical nature [3], and by evolutionary information [4]. 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* (patch for short in the sequel) 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 [1]. 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 [5]. 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 [6]. This model was also used to show that conserved residues tend to locate in the core [7]. In a more biological perspective, double mutant cycles were used to evidence the modular structure of binding patches [8]. 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 [9] and [10], the patch model consists of picking the k-nearest exposed neighbors of a central atom, resulting in disk-shaped (i.e. isotropic) patches.

Comparing patches. The design of structural comparisons and alignment tools has been carried out with two privileged applications. The first one is the analysis of bound complexes and the classification of their interfaces. In [11], 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 [12]. These analyses, which are concerned with the specificity of biological interactions, ultimately aim at inferring whether two proteins interact, using information from interacting homologous proteins. 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 [13], or particular interest are the methods inferring similar patches from the proteins' exposed surface. In particular, in [14], similar patches are inferred by merging small graphs (graphlets) whose $RMSD_d$ is upper-bounded, and which exhibit comparable bio-chemical properties.

Predicting binding affinities. The prediction of binding affinities is also a critical endeavor, as reliable predictions would have a major impact on applications to protein engineering and the study of interactomes [15]. The size of an interface, measured by the buried surface area ΔASA upon complex formation—or the number of interface atoms since both quantities are known to be proportional [6, 16], is known to correlate to the binding affinity, at least for rigid docking cases [17]. Intuitively, the size hints at the complex stability [5], as small interfaces may favor a fast turn-over of the interactions (e.g. in the enzyme - substrate case), while larger interfaces may favor more stable interactions (e.g. in the enzyme - inhibitor case). In addition to predictions based on such simple structural descriptors, a variety of scoring functions have been developed, their power to single out native-like complexes being assessed within the CAPRI experiment [18, 19]. Their ability to estimate binding affinity was also recently evaluated in the context of the binding affinity benchmark [15, 17].

1.2 Contributions

Questions addressed. As just recalled, a variety of geometric models have been proposed to investigate specific biological - biophysical properties of binding patches, and to compare binding patches. Following our analysis of protein interfaces, where enhanced geometric resulted in refined insights on biophysical properties [20, 21, 22], we undertook the task of designing a hierarchical binding patch model amenable to a variety of structural studies. More specifically, the patch model introduced herein was designed to address the following questions.

Dissecting the morphology of patches. As evidenced by previous work, the morphology of molecular shapes has essentially been described in geometric terms, using the atomic positions. On the other hand, in mathematics, topology is prosaically defined as the geometry of rubber made objects. In other words, topological features can be conserved, while geometric features are not. Our goal has therefore been to develop a patch model accommodating both the geometry and the topology of patches. Conceptually, enumerating the binding morphologies used by Nature is appealing. Application-wise, such morphologies are especially interesting wherever (isotropic) patch models are used, in particular to seek putative patches on orphan proteins.

Comparing patches. Taking for granted a patch model encoding both geometric and topological features, our second goal has been to design comparison algorithms operating at the atomic level, and privileging either features. Geometrically, the ability to compare patches at the atomic level is of interest to search similar patches, and in the context of structure-specificity studies, to assess if similar patches are involved in similar biological functions. Also, the possibility to assess geometric versus topological deformations undergone during docking is of interest to assess the difficulty of docking problems. One indeed expects a rising difficulty, from geometrically rigid partners to fully flexible partners, with topologically rigid (but geometrically flexible) cases as a third tier.

Identifying patches and specificity analysis. The specificity of protein interactions relates to the ability of a protein to bind selected partners, and a key problem in post-genomic studies consists of pulling back functional annotations from a given protein to homologous proteins. Conservation information can be used to do so, but such approaches are unable to handle the cases where similar patches are accounted for by different amino-acids. One of our goals has therefore been to investigate the ability of our patch model to detect a specificity related signal amidst a database of biological complexes organized into biological families.

Predicting binding affinities. Finally, a key strength of a patch model lies in its ability to support binding affinities estimations, at least in selected cases. We investigated this ability using the binding affinity benchmark.

Methodological positioning. Our patch model supports the questions just discussed thanks to its multi-scale structure. In a nutshell, we use the topological incidences of the spherical caps making up the solvent accessible surface of the interface atoms to define an integer-valued depth of these atoms, which goes beyond the classical core-rim model. Next, we assign atoms of identical depth into so-called *shells*, and use the relative position of these shells to define a graph coding the topology of the patch—the geometry of a shell pertaining to its associated atoms. This encoding is used to solve a relaxed structural alignment problem through a maximum clique calculation—a NP-hard problem, as initially proposed in [23]. 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

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

This problem has long been known to be equivalent to the maximum clique problem in the so-called product graphs [23, 24]. The maximum clique is a well known NP-Hard problem [25] that can be tackled with enumeration algorithms [26, 27] or optimization algorithms [28, 29, 24]. Given that the size of the product graphs is quadratic in the number of atoms, and that typical patches involve from 100 to 500 atoms, the product graphs have a number of vertices beyond tens of thousands (up to 143385 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 algorithm to shells. As we shall see, the problems solved still challenge the state-of-the-art maximum clique algorithms.

2 Methods

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*; 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 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 Fig. 1 and Fig. SI-1. We now sketch the main steps of the atom shelling tree construction, and refer the reader to the supplemental section 6.1 for the details.

Defining patches. We identify the interface atoms with our Voronoi interface model [20, 21], whose definition and software are presented in [30] and [22]. 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 is encoded in a so-called Half-edge Data Structure (HDS) [31], that gives access to the faces, 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 $SO(s) + 1 = SO(t)$: 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 [32]. 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 6.2.1, 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 two strategies to compare the topology and the geometry of patches, respectively.

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 $\min(\mid s_1 \mid, \mid s_2 \mid)$ atoms in-between the two shells, or equivalently, to a cost $\max(\mid s_1 \mid, \mid s_2 \mid) - \min(\mid s_1 \mid, \mid s_2 \mid)$. 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 patches, the TED cost is the following symmetric difference

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

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 two patches, 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 6.2.2). 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 Identifying Patches and Specificity 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 family by distinguishing the ligand and the receptor of each complex. Thus, the database of patches is partitioned into *typed families.* 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 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 family classification, and the possibility of using the dissimilarity score for generating automatic classifications of patches. It is also related to the morphology - specificity relationship, since the proteins in this case meet two criteria, namely they bind the same family of ligand, and their binding patches possess similar morphologies. 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. This information is of particular interest for docking, where the steric complementarity between the partners is often used as a matching criterion. Case III highlights contradictions between the typed family classification and the dissimilarity score values.

3 Results

3.1 Database

The results presented in this section were obtained on a database of 498 patches generated from 249 complexes. These complexes are heterodimers and one of our goals is to study the symmetry of patches. The set of complexes was assembled from two sources. The first one is the IMGT 3D structure database [33], from which we extracted 116 immunoglobulin - ligand structure, with resolution ≤ 2.0. The second one is the recently assembled binding affinity benchmark [17], a manually curated dataset involving 133 complexes with experimentally measured binding affinity, and resolutions in the range $1.1 - 3.3$, the median being 2.4. (In fact, the affinity benchmark contains 144 complexes, but 11 were redundant with the ones that we extracted from the IMGT 3D database.) The reader is referred to the Tables SI-1 and SI-2 for the presentation of the typed families and for the exhaustive list of PDB ids.

The rationale in assembling this database has been the following. First, all the complexes are known to be biological complexes—as opposed to crystallization artifacts. Second, the database strikes a balance in terms of diversity and homogeneity: on the one hand, it involves a large variety of complexes and biological functions, in particular the so-called O set of the affinity benchmark; on the other hand, the subset from IMGT 3D gives the opportunity to compare a significant number of proteins involved in the same biological function. Third, the affinity benchmark allows assessing the geometrical and topological changes undergone by the patches upon binding, since the bound and unbound forms of each partner are known.

We used this set of complexes for two purposes. On the one hand, the morphological study of section 3.2 was carried out on the whole database. On the other hand, the clustering and identification study of section 3.4 was performed on the subset consisting of complexes involving enzymes or immunoglobulins. (The variety of complexes involved in the aforementioned O set precludes the identification of patches with significant similarity.)

All the patches but two (496 over 498) possess between 14 to 294 atoms and 1 to 20 shells. The two exceptions are the patches of the signaling complex 2oza, since chain B contributes 395 atoms and 28 shells (Fig. SI-4), while chain A contributes 365 atoms and 47 shells (Fig. SI-5). Also, the 498 patches yield a total of $\binom{498}{2} + 498 = 124251$ pairwise comparisons.

3.2 Dissecting the Morphology of Patches

Canonical Morphologies. 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 (Fig. 2). Since this plot exhibits a continuous variation, we extract typical morphologies by examining extreme cases for a fixed number of shells and atoms, respectively. Minimizing and maximizing the number of atoms for a fixed number of shells yields *tubular* and *pyramidal* shapes (Fig. 3). Similarly, minimizing and maximizing the number of shells 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 (data not shown) allows us to single out five cases (Fig. 4). A curve contained in a narrow horizontal slab corresponds to a tubular shape. A curve involving an increasing section followed by a decreasing section corresponds to a pear-like shape. In the remaining cases, the curve is decreasing. The maximum SO may be used to distinguish isotropic (large max SO) from anisotropic (small max SO) patches. As illustrated on Fig. 3, in each case, the geometry can be used to define flat versus non-flat (pyramidal) patches.

Asymmetry Between Partner Patches. 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 atom shelling 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 periphery. 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 (Figs. 5 and Fig. 6).

3.3 Comparing Patches

3.3.1 Efficacy of the Atom Shelling Tree Encoding

Given the hardness of the geometric matching problem, as specified by Eq. (1), we first report the running times of the topological and geometric comparisons, whose punchline consists of localizing calculations at the shell level.

On a 100 nodes cluster equipped with Intel Xeon processors at 2.66Ghz, the 124251 pairwise comparisons of our database took 622 seconds with TED_t, 41901 seconds with TED_g ($\epsilon = 1$Å), and 1076390 seconds with TED_g ($\epsilon = 2\text{Å}$). For the latter, a time limit of 2 hours per instances was used, and 13 instances remained unsolved.

To compare the topological and geometric matchings, we plot DIS_g against DIS_t. While it has been observed in section 2.2 that $\text{DIS}_g \geq \text{DIS}_t$, Fig. 7 illustrates the variation DIS_g as a function of DIS_t. Of particular interest are pairs of patches with a similar topology (low DIS_t) but different geometries (high DIS_g). Two such patches, having respectively 82% and 32% of common atoms from the topological and geometric standpoints (Fig. SI-6).

Finally, we assess the accuracy of our geometric matchings, an important issue since 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. The $RMSD_d$ globally increases with the geometric dissimilarity, and so does the variance of the $RMSD_d$ for a fixed DIS_g (Fig. SI-7). Interestingly, the $RMSD_d$ remains moderate (less than 4Å) for a geometric dissimilarity smaller or equal to 0.25 (i.e. between patches having at least 75% of geometrically common atoms), showing that seeking quasi-isometric subsets at the shell level is effective to compare whole patches. Note that from now on, we assume that two patches are geometrically (resp. topologically) similar if their DIS_g (resp. DIS_t) is less or equal to 0.25.

3.3.2 Patches and their Preimages

Computing preimages. The topological similarity being less stringent than the geometric one, we use it to scale the conformational changes undergone by the partners of the binding affinity benchmark. To this end, consider a partner of a complex, in its bound and unbound forms: the patch being defined from the bound partner, we define the corresponding atoms on the unbound partner as the patch *preimage*.

Preimages were generated for the complexes coming from the full Affinity benchmark (144 complexes) as follows. First, a one-to-one chain mapping between the chains of the bound and unbound partners was sought, and retained in case of sequence identity higher than 90% for any two chains put in correspondence. (Alignments were generated exactly i.e. without any substitution matrix.) These criteria dismissed 43 complexes. Second, for the 101 valid complexes, we produced atom mappings between the residues matched by the sequence alignments. Then, we generated the preimage, requiring the presence on the preimage of at least 90% of the patch's atoms. Over the 202 patches, only 126 met this requirement.

On docking difficulties. For each pair (patch, preimage), we plotted the topological dissimilarity against the geometrical dissimilarity (Fig. 8). While this plot exhibits a continuous distribution and a monotonic correlation (Spearman coefficient of 0.87), three cases can be singled out (Fig. 9). Two extreme cases correspond to rigid and flexible docking, where both the topology and the geometry hardly change and significantly change, respectively. The intermediate situation arises when the patch and its preimage have a similar topology but a different geometry: such a patch is called *topo-rigid*. In contrast to our analysis, with an interface RMSD (I-RMSD) of 0.17Å, 0.48Å, and 0.35Å respectively, the rigid, topo-rigid and flexible patches appear as rigid from the alpha carbon point of view [17].

On morphologies and docking predictions. In a second experiment, we tested whether similar preimages lead to similar patches upon binding (Fig. 10(A)). For any two receptor and any two ligand patches, we plotted the topological dissimilarity between preimages against the topological dissimilarity of the resulting patches (Fig. 10(B)), and the geometrical dissimilarity between preimages against the geometrical dissimilarity of the resulting patches (Fig. 10(C)). The corresponding Pearson correlation coefficients are respectively of 0.69 and 0.91, with a p-value smaller than $1e^{-99}$ in both cases. Interestingly, topologically similar preimages ($\mathrm{DIS}_t \leq 0.25$) yield topologically similar patches in 88% of the cases (over the 1238 pairs of similar preimages, 1093 lead to similar patches). On the other hand, we could not verify whether geometrically similar preimages produce geometrically similar patches, due to the paucity of geometrically similar preimages in our dataset (see Computing preimages, above).

3.4 Identifying Patches and Specificity Analysis

The identification results, as specified in section 2.3, are summarized in Table 1 for TED_t and TED_g, with different thresholds for the latter parameter. For all methods, a large discrepancy is observed between cases I and II, further stressing the lack of symmetry of patches. Moreover, immunoglobulin and enzyme typed-families exhibit extreme behaviors (Table 1 and Table SI-3). About 70% of the immunoglobulin patches have their nearest neighbor coming from their own biological family, showing that proteins binding the same type of ligand possess a similar morphology.

To further understand the different identification rates of typed families, we proceed as follows. First, we compute the Wilcoxon-Mann-Whitney rank-sum test of $s(P, P)$ against $s(P, P^c)$, also known as the U test, the null hypothesis being that the two series of scores come from the same distribution. Second, we compute the Spearman's correlation coefficient between the negative log of the U-test and the identification rate, as this test aims at detecting monotonic correlations. The correlation coefficient obtained, equal to -0.893, shows the strong coupling between the family consistency (measured by the U test) and the identification rate. In other words, on consistent families, our comparison achieves good identification rate (Fig. SI-8).

3.5 Predicting Binding Affinities

Binding affinities versus structural parameters. The binding affinity of a protein complex can be described by its equilibrium dissociation constant K_d, which is related to the dissociation Gibbs free energy by $\Delta G = -RT \ln K_d/c^\circ$ (in the $c^\circ = 1\ M$ standard state). We thus investigate the correlations between $-\ln K_d$ and our descriptors. In addition to the number of shells and the depth of a binding patch, we define its *internal path length* as $\mathrm{IPL} = \#\mathrm{atoms} \times \overline{SO}$. Equivalently, the IPL is the sum for all the atoms of their depth in the atom shelling tree, the root being at depth zero. From a statistical standpoint, we computed three correlations coefficients: Pearson's coefficient $\mathrm{C_{Pea}}$ which aims at detecting linear correlations, Spearman's coefficient $\mathrm{C_{Spe}}$ which is better suited to detect non affine monotonic correlations, and the recently designed Maximal Information Coefficient $\mathrm{C_{MIC}}$ [34]. The latter remarkably targets general functional associations, and also behaves consistently across functional associations plagued with the same noise level. P-values based on permutation tests were also computed. (Most of the p-values of the MIC coefficient are missing, though, since the tables delivered with the Supporting Information of [34] do not feature p-values for small correlations: e.g., no value is provided for $\mathrm{C_{MIC}} \leq 0.27$ in the table of size $n = 140$.)

We computed these correlations at the database level (Table SI-5), and also for the three classes of low, moderate and high flexibility introduced in [17] (Tables SI-6 and SI-7). The IPL consistently yields the best correlation, both in terms of correlation coefficient and p-value, the contenders being the number of atoms and the interface depth, and to a lesser extent the interface area ΔASA. For example, $\mathrm{C_{Spe}}(-\ln K_d, \mathrm{IPL}) = -0.43$ and $\mathrm{C_{Spe}}(-\ln K_d, \#atoms) = -0.37$, while $\mathrm{C_{MIC}}(-\ln K_d, \mathrm{IPL}) = 0.35$ and $\mathrm{C_{MIC}}(-\ln K_d, \#atoms) = 0.24$. If the correlations coefficients are moderate, their significance is strong. Correlations computed on classes of varying flexibility yield better results, in particular for the class I-RMSD ≤ 1. With $\mathrm{C_{Spe}}(-\ln K_d, \mathrm{IPL}) =$

-0.59 versus $\mathrm{C}_{\mathrm{Spe}}(-\ln K_d, \#atoms) = -0.58$, and $\mathrm{C}_{\mathrm{MIC}}(-\ln K_d, \mathrm{IPL}) = -0.48$ versus $\mathrm{C}_{\mathrm{MIC}}(-\ln K_d, \#atoms) = -0.43$, the IPL still provides the best correlation.

Binding affinities versus structural changes. A question often faced consists of understanding why related complexes exhibit very different affinities. This issue is discussed in detail for nine pairs of complexes in [17], where it is noted that the two complexes of a pair have a *similar geometry.* Using our encoding of patches, we computed the geometric dissimilarity of the receptor and ligand patches (Table SI-8).

Except for complexes 2ptc__E-I and 2tgp__Z-I, the fact that $\mathrm{DIS}_g \geq 0.35$ shows that the receptors are actually geometrically different, and likewise for the ligands, despite seemingly modest changes in amino-acid sequence. Indeed, in the case of 2vir__AB-C vs 2vis__AB-C, only one residue substitution in the receptor side occurs, leading to geometric dissimilarities of 0.35 and 0.23 between the receptors' and ligands' patches, respectively; for 1efn__B-A vs 1avz__B-C, a one residue substitution in the ligand side results in geometric dissimilarities of 0.45 and 0.48 between the receptors' and ligands' patches, respectively.

4 Discussion

4.1 Dissecting the morphology of patches

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, $\Delta\Delta G$) [5, 6, 35, 16, 7]. Selected such parameters have also been traced along molecular dynamics [36]. While trends have emerged for collections of complexes [36, 7, 1], conclusions from meta-analysis, when refined under the lens of sharper structural parameters, may not apply to isolated complexes [21]. 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 [37, 9, 10]. 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 elaborates on our Voronoi interface model [20, 21, 30], which offers a unified way to refine classical interface parameters [1], and proved instrumental to transpose conclusions from the database level to the single complex level, regarding the biochemical properties of interfaces [6], the geometry of conservation [7], as well as the solvation of residues along molecular dynamics trajectories [36].

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. These typical morphologies show that the isotropic patch models used so far do not account for the variety of morphologies encountered. Phrased differently, using isotropic patch models is a hindrance to identify potential binding patches. Our encoding also allows assessing the symmetry of the two patches of a complex, either directly from the dissimilarity score, or indirectly based on statistics derived from the atom shelling tree—the average shelling order $\overline{SO}$. In particular, we believe that the ability to evidence the lack of symmetry for selected biological families will shed new light on the problem of determining surface complementarity for rigid body docking.

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 [9, 10]. 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 bio-chemical compatibility between the atoms matched.

4.2 Comparing patches: geometry versus topology

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 [11, 38], 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 [12] 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 [14], 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 accommodate independently topological and geometric comparisons, based on the atom shelling tree encoding. We have seen that the former comparison if more lenient than the latter, and that both benefits from the atom shelling tree encoding to perform the comparison at the shell rather than whole patch level. Moreover, algorithm TED_g is accurate to perform atomic scale comparisons, since a geometric dissimilarity $\text{DIS}_g < 0.25$ yields $RMSD\_d < 4$.

Second, the ability to perform the matching at the atomic level, as opposed to the residue and functional group levels in [11, 12], motivated the study of patches and their preimages on the binding affinity benchmark. We have seen in particular that cases which exhibit comparable difficulty level in terms of interface RMSD actually correspond to different cases, namely rigid, flexible, and topo-rigid. This latter case is actually somewhat expected, as in the conformer selection model [39], the presence of a topologically pre-formed patch may ease the formation of the encounter complex, and thus of the final complex. Knowing that a patch is topo-rigid is also of interest for sampling the conformational space, as the extra constraints imposed by the stable topology may enable the sampling of a sub-space of the whole conformational space of that partner. In the same spirit, our topological comparison should also prove useful to assess docking results, as done in the CAPRI experiment, to complement the purely geometric criteria currently used [18]. Another striking point outlined by our analysis is the stability of the topological signature of the patches upon docking, since we observed that 88% is similar preimages are resulting in similar bound patches. This information is clearly of high interest for template-based docking, to pull-back information on a putative bound partner, from the bound structure of a similar patch.

4.3 Identifying patches and specificity analysis

The coherence analysis of typed families, as revealed by the identification rates and the clustering properties, shows that the atomic level geometric and topological comparisons provide stringent criteria. In particular, algorithm TED_t being more lenient than algorithm TED_g, the topological consistency is better than the geometric one. However, diverse typed families such as the so-called O set of the affinity benchmark currently appear as very diverse. The identification rates also shed light on the morphology - specificity relationship, since they highlight the ability of patches of a family of receptors to bind a specific ligand. The fact that 70% of the immunoglobulin patches have their nearest neighbor coming from their own biological family (Table 1) is interesting in two respects. First, this shows that immunoglobulin patches that binds the same kind of ligand possess similar patch morphologies (i.e. low dissimilarity scores), while immunoglobulin patches that binds different kinds of ligand (e.g. a peptide versus a small chemical) possess different morphologies (i.e high dissimilarity scores). Second, the characteristics of immunoglobulin patches are different from those of the 122 enzyme-related patches coming from the Affinity benchmark.

These specificities call for further developments. In the clustering of interfaces and patches of [11, 38], class III clusters gather geometrically similar patches involved in dissimilar interfaces, i.e. patches having more than one binding function. In a related vein, the specificity of interactions has been studied in [40] using 3D templates pre-processed into a database of interaction types based on PFAM domains, together with information on the conservation of residues. 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. More generally, 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 [41]. 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 [42], which assigns a unique numbering to the amino-acids of the complementarity determining regions (CDR). The CDR are responsible for the specificity of the immune response in general and for affinity maturation in particular. Structural modeling of this latter phenomenon is especially challenging since conservation information is irrelevant, and a unique mutation may result in a significant change of the affinity [43]. We foresee that our patch model will prove useful both to assess the geometric changes associated to mutations—as reported herein for seemingly geometrically similar complexes of the affinity benchmark, and also to understand with canonical morphologies account for specific interactions for a given family of antigens.

4.4 Predicting binding affinities

Dissociation constants of biological complexes are known to span the range $10^{-5} < K_d < 10^{-14}$, and are also known to depend on factors such as the temperature, the ionic strength, molecular crowding, and in particular the pH [17]. Because the free energy change ΔG has enthalpic and entropic terms, estimating the binding affinity from features of the bound complex is doomed to fail in the most general setting, as the entropic changes cannot be estimated—letting alone the entropic changes of the solvent, and complexes formed by (semi-)rigid docking seem more accessible [17].

Recently, the Pearson correlation coefficients between $-\ln K_d$ and various scores were reported in [15], on a database of 81 protein complexes. The best correlation obtained is that of the FIREDOCK score [44], with a value of $\mathrm{C_{Pea}} = -0.32$ and a p-value in the range $0.01 \leq$ p-val ≤ 0.05. On the whole affinity benchmark, we observe that the same correlation drops down to $\mathrm{C_{Pea}} = -0.17$, a value much worse than $\mathrm{C_{Pea}} = 0.31$ observed with our internal path length (Table SI-5).

While the latter correlation remains mediocre, the ability of our purely geometric and topological descriptor to outperform the complex top-scoring function of [15] is interesting in several respects. On the one hand, the IPL can be seen as a weighted version of the number of interface atoms, the weight of an atom being is distance from the rim of the binding patch. Given the expected correlation between the interface size (or the number of atoms) and the affinity, the gain provided by the encoding of the depth should not come as a surprise. The corresponding plots (Fig. SI-9) also show that minimum values of the parameters are required to reach a given high affinity. For example, for $K_d \leq 10^{-10}$ or equivalently $-\ln(K_d) \geq 23.02$, one observes that the minimum average number of atoms is 110, that the minimum average depth is 3 (i.e 4 concentric shells), that the minimum IPL is 160.5, and that $\Delta ASA \geq 1684^2$. In other words, the typical values of the parameters required to match the prescription $K_d \leq 10^{-10}$ correspond to the typical average values observed for protein-protein complexes [1, Table 2].

On more speculative grounds, the depth of atoms within a binding patch is likely related to dynamic properties, both of the partners and of the solvent. For the latter, it has also been observed in [21] that the Voronoi shelling order of an interface has a strong correlation with the dynamics of water molecules squeezed in-between the partners. For the partners, it is appealing to believe that the increase of internal entropy observed upon binding is related to the shape of the interface coded by the two shelling trees and their matching, since this matching precisely describes the coupling between the two sub-units. Developing quantitative models exploiting our encoding of depth might therefore prove fruitful for the prediction of dynamic properties and the improvement of binding affinity predictions, at least for selected classes of complexes.

References

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

[2] 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.

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

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

[5] 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.

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

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] P.L. Kastritis and A.M.J.J. Bonvin. Are scoring functions in protein- protein docking ready to predict interactomes? clues from a novel binding affinity benchmark. *Journal of proteome research*, 9(5):2216–2225, 2010.

[16] 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.

[17] P.L. Kastritis, I.H. Moal, H. Hwang, Z. Weng, P.A. Bates, A. Bonvin, and J. Janin. A structure-based benchmark for protein-protein binding affinity. *Protein Science*, 20:482–491, 2011.

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

[19] E. Feliu and B. Oliva. How different from random are docking predictions when ranked by scoring functions? *Proteins*, 78(16), 2010.

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

[21] 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.

[22] 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.

[23] 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.

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

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

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

[27] 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.

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

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

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

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

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

[33] F. Ehrenmann, Q. Kaas Q, and M-P. Lefranc. Imgt/3dstructure-db and imgt/domaingapalign: a database and a tool for immunoglobulins or antibodies, t cell receptors, mhc, igsf and mhcsf. *Nucl. Acids Res.*, 38:D301–307, 2010.

[34] David N. Reshef, Yakir A. Reshef, Hilary K. Finucane, Sharon R. Grossman, Gilean McVean, Peter J. Turnbaugh, Eric S. Lander, Michael Mitzenmacher, and Pardis C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.

[35] 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.

[36] 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.

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

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

[39] R. Grünberg, M. Nilges, and J. Leckner. Flexibility and conformational entropy in protein-protein binding. *Structure*, 14(4):683–693, 2006.

[40] A. Panjkovich and P. Aloy. Predicting protein–protein interaction specificity through the integration of three-dimensional structural information and the evolutionary record of protein domains. *Mol. BioSyst.*, 6(4):741–749, 2010.

[41] 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.

[42] 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.

[43] S.M. Lippow, K.D. Wittrup, and B. Tidor. Computational design of antibody-affinity improvement beyond in vivo maturation. *Nature biotechnology*, 25(10):1171–1176, 2007.

[44] E. Mashiach, D. Schneidman-Duhovny, N. Andrusier, R. Nussinov, and H.J. Wolfson. Firedock: a web server for fast interaction refinement in molecular docking. *Nucleic acids research*, 36(suppl 2):W229–W232, 2008.

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

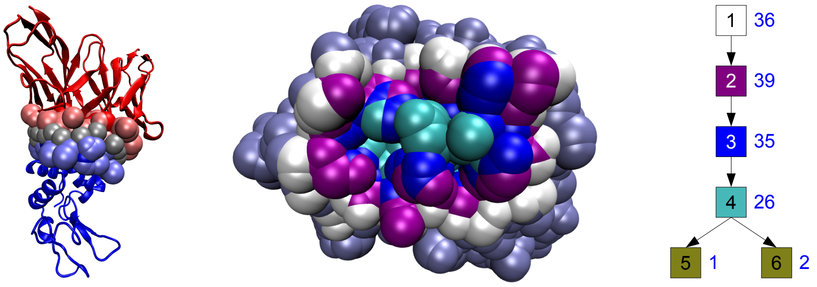
[46] P.M.M. De Castro, F. Cazals, S. Loriot, and M. Teillaud. Design of the cgal spherical kernel and application to arrangements of circles on a sphere. *Computational Geometry: Theory and Applications*, 42(6-7):536–550, 2009. Preliminary version as INRIA Tech report 6298.

[47] 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.

[48] 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.

5 Artwork

Figure 1 Shelling a patch: illustration. (a) Side view of the protein complex 1vfb, an immunoglobulin - antigen complex, with interface atoms displayed in red for partner A (chain A and B) and in blue for partner B (chain C). Gray atoms correspond to the interfacial water molecules. **(b,c)** Rotated view of the patch of partner A, shelled into concentric *shells* of atoms, 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 gray-blue, and the atoms of a shell are represented with a given color.



(a) (b) (c)

Table 1 Identification rates via the typed 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). The values between parentheses present the identification rates obtained when p belong to one of the immunoglobulin typed-families (first value) or to one of the enzyme typed-families (second value).

Method	case I: $\hat{p} \in P - \{p\}$	case II: $\hat{p} \in \overline{P}$	case III: $\hat{p} \in P^c$
TED$_t$	41.6% (50.4%, 23.8%)	9.8% (6.5%, 13.4%)	48.6% (43.1%, 62.8%)
TED$_g$: $\epsilon = 1$Å	57.6% (69.5%, 33.6%)	3.3% (1.6%, 6.6%)	39.1% (28.9%, 59.8%)
TED$_g$: $\epsilon = 2$Å	58.1% (70.3%, 33.6%)	5.2% (2.8%, 9.8%)	36.7% (26.9%, 56.6%)

Figure 2 Morphology of the patches: number of atoms versus number of shells. The bold symbols identify the canonical morphologies presented on Fig. 3. For the sake of readability, the displayed range for the number of shells is [0, 20], so that the figure does not show the patches of the signaling complex 2oza (chain A: 26 shells and 395; chain B: 47 shells and 365 atoms).

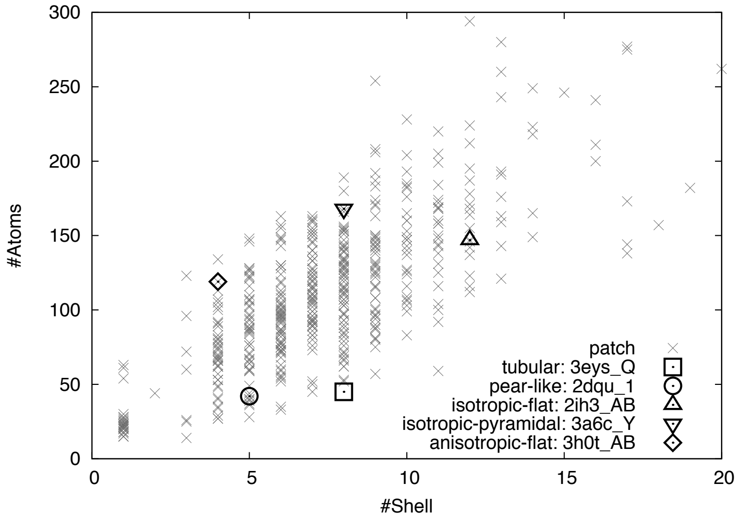
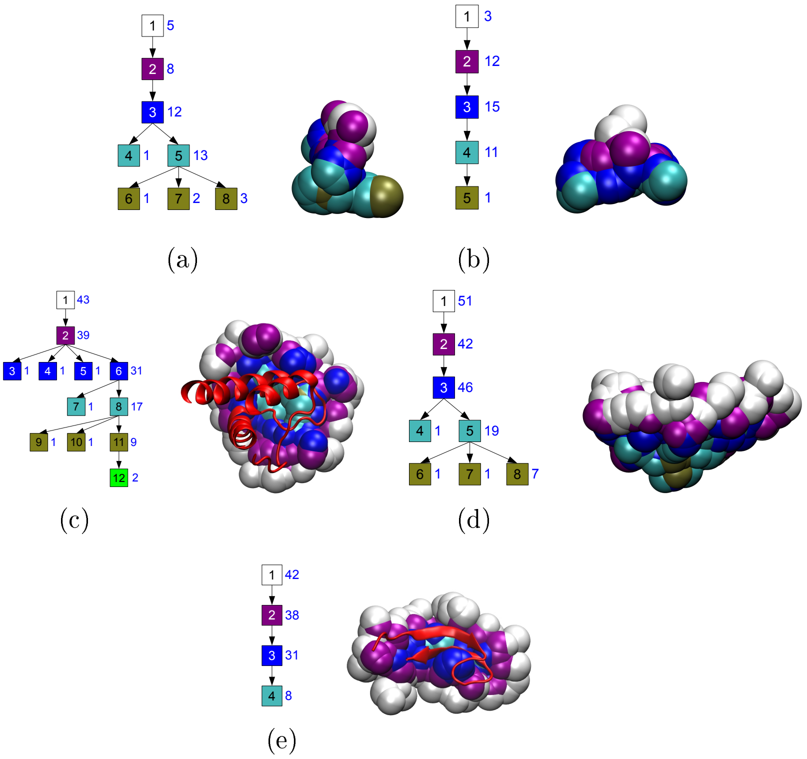


Figure 3 Illustration of the five canonical morphologies. The ligands are represented as cartoons when they do not clutter the picture. **(a)** tubular (pdbid 3eys, chain Q) **(b)** pear-like (pdbid 2dqu, chain 1) **(c)** isotropic-flat (pdbid 2ih3, chains AB) **(d)** isotropic-pyramidal (pdbid 3a6c, chain Y) **(e)** anisotropic-flat (pdbid 3h0t, chains AB)



(a) (b)

(c) (d)

(e)

Figure 4 Signature of the five morphologies of Fig. 3. The variation of the number of atoms at each SO, the maximum SO observed, and the geometry of the patch can be used to define canonical morphologies. The curves displayed correspond to the tubular, pear-like, isotropic and anisotropic morphologies of Fig. 3.

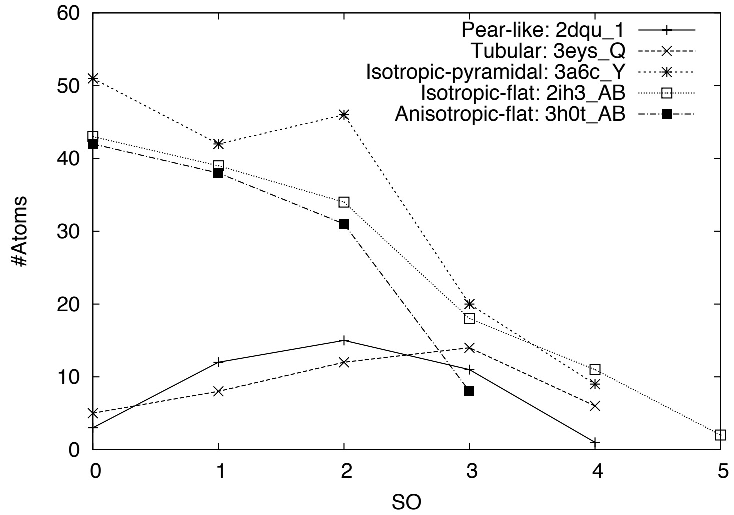


Figure 5 The asymmetry of the partner patches revealed by their average shelling order ($\overline{SO}$). For each typed family of immunoglobulin complexes and of enzyme complexes, the minimum, average and maximum $\overline{SO}$ of the receptor and ligand of a family are plotted next to one-another. The largest discrepancy is observed between the $\overline{SO}$ values of the AA_Pept_L and AA_Pept_R classes.

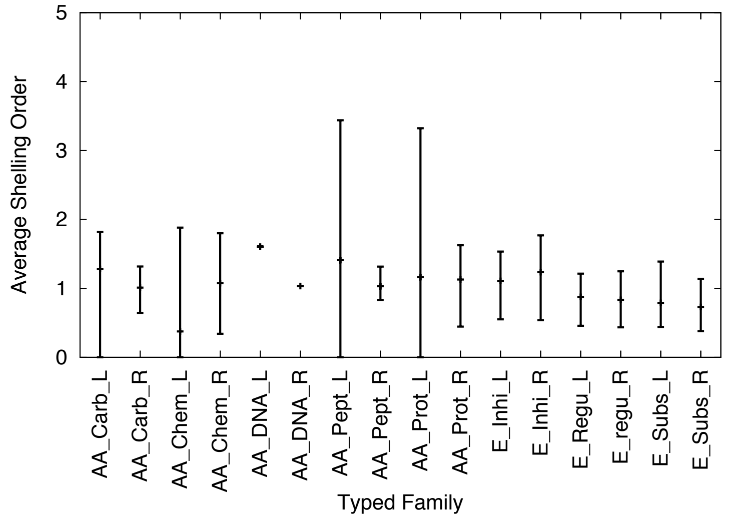


Figure 6 An example complex whose asymmetry is revealed by the average Shelling Order. Complex 3ifl, from A_Pept. Left: the receptor (chains HL). Non interface atoms are displayed in grey, and ligand is displayed with a cartoon representation. Right: the ligand (chain P) alone. The SO of the receptor vary in the range $[0, 4]$, with an $\overline{SO}$ of 1.17, while for the ligand the SO vary in range $[0, 7]$ with an $\overline{SO}$ of 3.47.

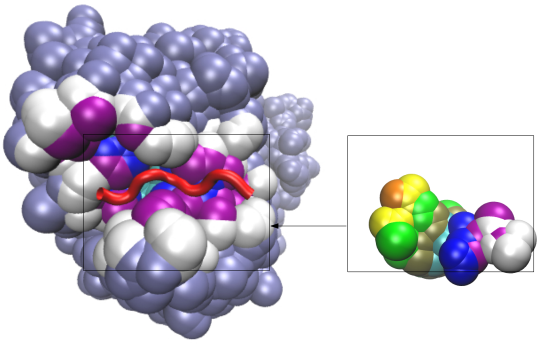


Figure 7 Topological (DIS$_t$) versus geometrical (DIS$_g$) dissimilarities. With one point per pairwise comparison, the plot illustrates the fact that the topological dissimilarity is a lower-bound of the geometrical dissimilarity. The black rectangle singles out instances having similar topologies (low DIS$_t$) but different geometries (high DIS$_g$).

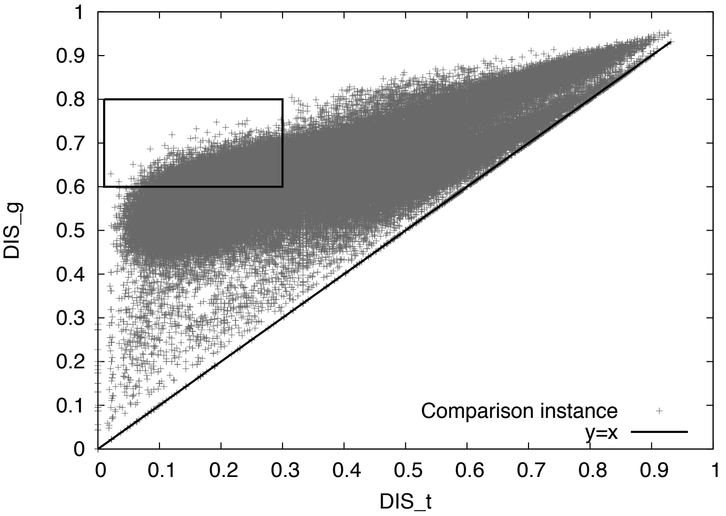


Figure 8 Topological versus geometric dissimilarities for pairs (patch, patch preimage on the unbound partner) of the binding affinity benchmark. The square and circle correspond to rigid and fully flexible docking, respectively. The triangle corresponds to a patch whose geometry but not topology changes upon binding: such a patch, called *topo-rigid*, is *preformed* on the unbound partner. The three highlighted examples are presented in figure 9

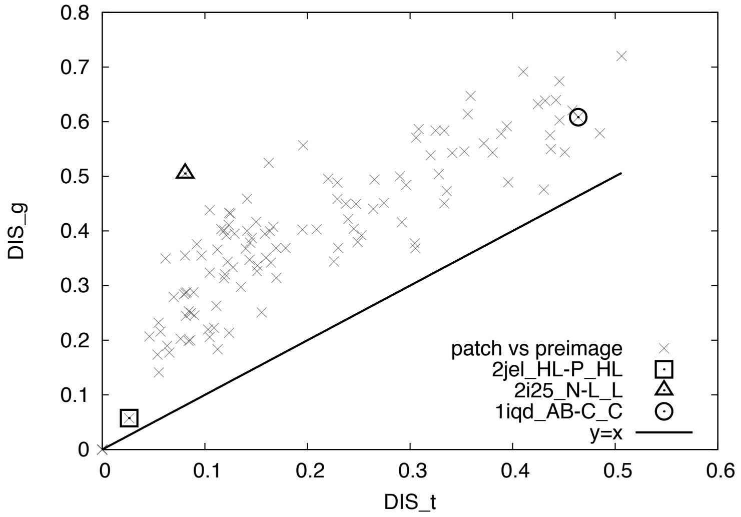


Figure 9 Rigid, topo-rigid, and flexible patches: the three cases singled out on Fig. 8. On each row, the left and right columns respectively display the patch preimage and the patch. Top: for complex 2jel_HL-P, the chains HL define a rigid patch, whose topology and geometry are preserved between the unbound and bound form ($\mathrm{DIS}_t = 0.026$, and $\mathrm{DIS}_g = 0.058$, associated with a $RMSD_d$ of 0.90Å). Middle: for complex 2i25_N-L, the chain L defines a topologically-rigid patch whose topology is preserved, but whose geometry is not ($\mathrm{DIS}_t = 0.081$, and $\mathrm{DIS}_g = 0.505$, associated with a $RMSD_d$ of 4.42Å). Bottom: for complex 1iqd_AB-C, the chain C defines a flexible patch whose topology and geometry undergo significant changes ($\mathrm{DIS}_t = 0.464$, and $\mathrm{DIS}_g = 0.608$, associated with a $RMSD_d$ of 2.46Å).

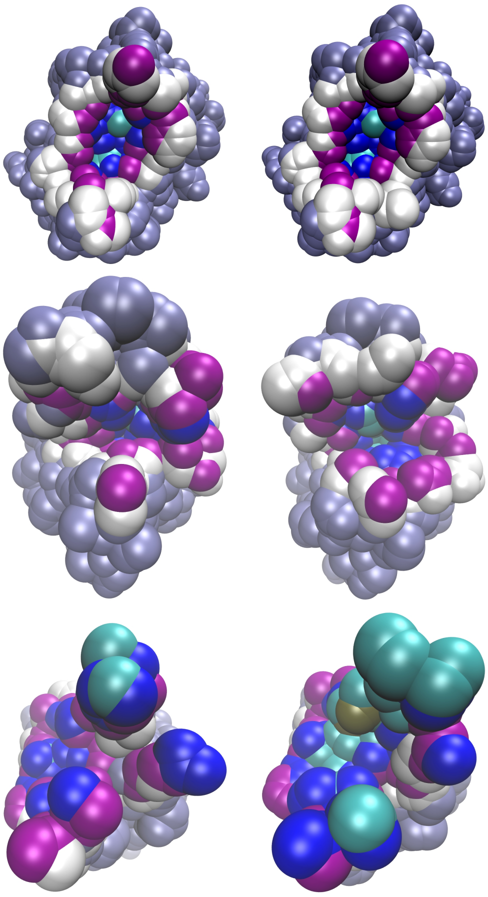
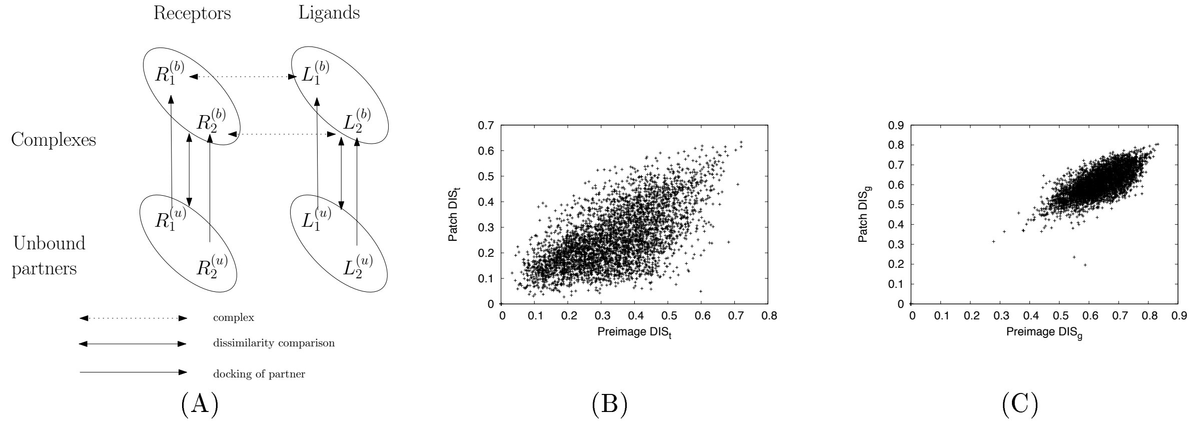


Figure 10 On morphologies and docking predictions. (A) Two complexes yield two comparisons between patches and their preimages, namely $DIS_x(R_1^{(b)}, R_2^{(b)})$ versus $DIS_x(R_1^{(u)}, R_2^{(u)})$ and $DIS_x(L_1^{(b)}, L_2^{(b)})$ versus $DIS_x(L_1^{(u)}, L_2^{(u)})$, where DIS_x refers to the topological or geometric dissimilarity. (B) Topological dissimilarity (C) Geometric dissimilarity.



6 Supporting Information

6.1 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. SI-1.

6.1.1 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 [20, 21, 30]. 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 [45], stored in a half-edge data structure (HDS) [31]. 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 [46].

6.1.2 Selected Properties of Patches

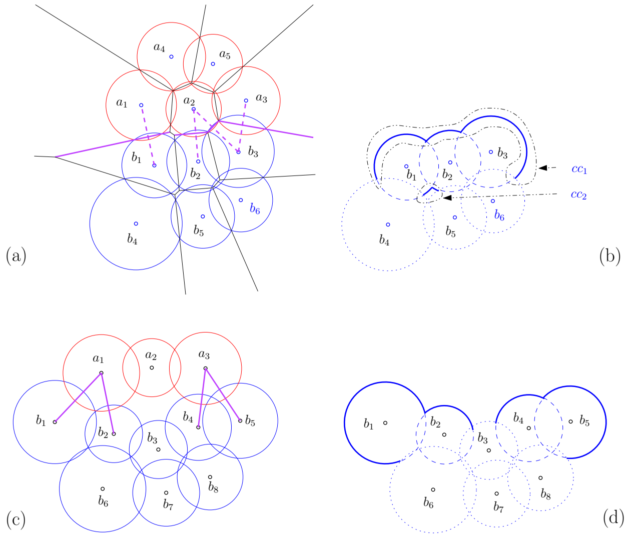
Before presenting the details of the topological encoding sketched in section 2.1, we discuss selected properties of patches.

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. SI-1(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 [16]. But as illustrated on Fig. SI-1(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).

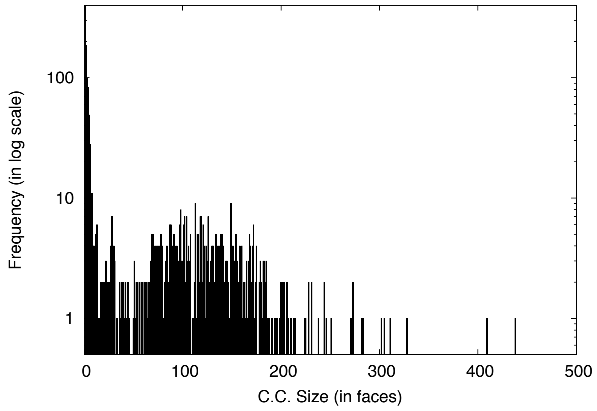
Supporting Information Figure 1 Defining and shelling a patch: 2D illustration. **(a)** The endpoint of the dashed purple edges, which are dual of the solid purple edges of the Voronoi diagram, identify the interface atoms. **(b)** The patch of the blue subunit is the Solvent Accessible Surface of its 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, namely the outer one, only is used to initiate the shelling order calculation.



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 the Fig. SI-2, which has an empty gap between the range [1,13] and [15,438].

Practically, having computed the connected component of the patch, we remove all components containing less than 15 faces— results in the removal of at most 15 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*.

Supporting Information Figure 2 Frequencies of connected component of given size. Computed over the 498 shelling graphs generated from our dataset, the histogram presents an empty gap between the range [1,13] and [15,438].



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 [20] and explained in [22], 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 [20].

6.1.3 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 [45]. A certified embedding in 3D is obtained thanks to the robust geometric operations described in [46].

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 [47], 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.

6.2 Comparing Patches: a Generic Dynamic Programming Based Approach

6.2.1 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 [32]. 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. Indeed, removing the root of an ordered tree leaves a forest of ordered 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 [48] has cubic time complexity, and quadratic memory requirements. Note that the TED problem for unordered trees is in general NP-hard [32].

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.

6.2.2 Comparing Shells

Our strategy to compare shells is a modification of the one proposed in [24] 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:

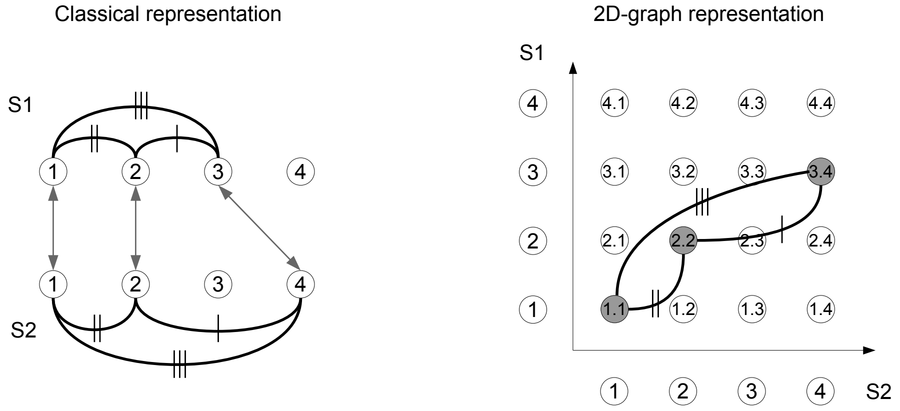
1. $i \neq j$ and $k \neq l$; this constraints ensures that an atom of s_1 can be matched with at most one atom of s_2, and vice versa;

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 (Fig. SI-3).

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 [25], and it has been studied extensively in literature. Interested readers can refer to [28] 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 [29].

Supporting Information Figure 3 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.



6.3 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 124251 pairwise comparisons of our database was done in about 622 seconds by TED$_t$, in about 41901 seconds by TED$_g$ ($\epsilon = 1$Å), and 1076390 seconds by TED$_g$ ($\epsilon = 2$Å). For the later, a time limit of 2 hours per instances was used, and 13 instances remained unsolved. 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.

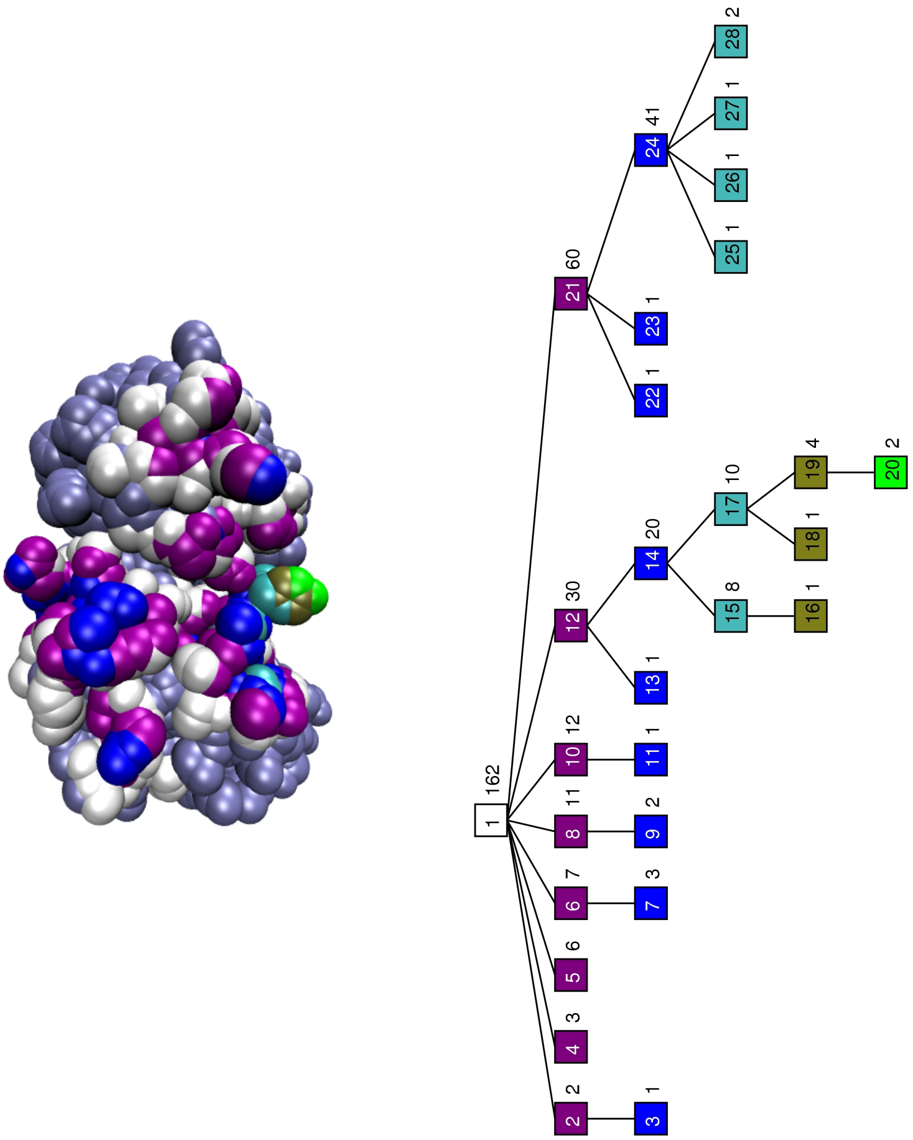
6.4 Dataset of Biological Complexes

Our approach is validated on a dataset of 498 patches generated from 249 complexes. The set of complexes was assembled from two sources. The first one is the IMGT 3D structure database [33], from which we extracted 116 immunoglobulin - ligand structure, with resolution ≤ 2.0. The second one is the recently assembled affinity benchmark [17], a manually curated dataset involving 133 complexes with experimentally measured binding affinity, and resolutions in the range $1.1 - 3.3$, the median being 2.4. (In fact, the affinity benchmark contains 144 complexes, but 11 were redundant with the ones that we extracted from the IMGT 3D database.)

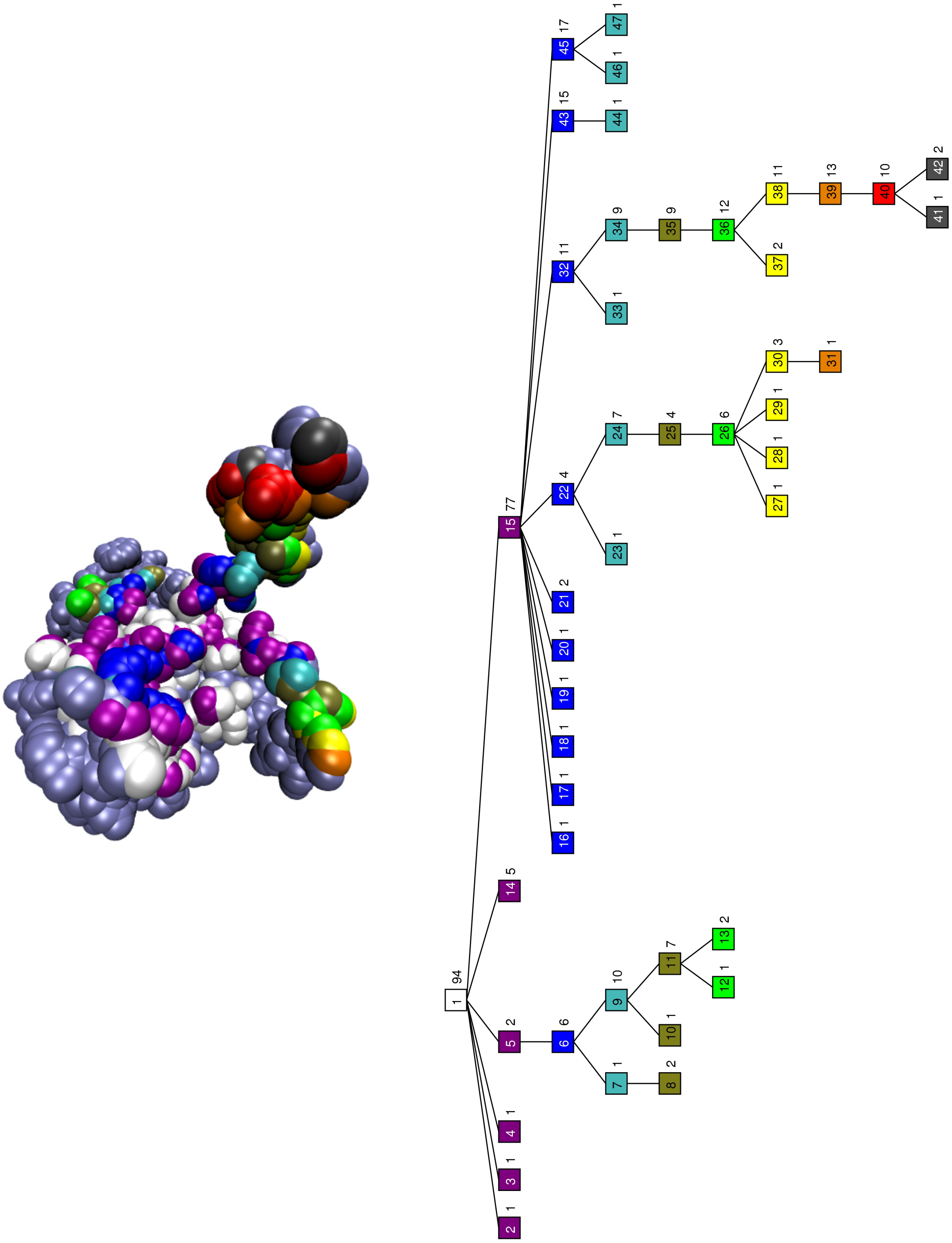
Note that while most of the patches possess between 14 to 294 atoms and 1 to 20 shells, the two patches from the signaling complex 2oza are larger: 2oza chain B possesses 395 atoms and 28 shell (Fig. SI-4), and 2oza chain A possesses 365 atoms and 47 shells (Fig. SI-5).

By distinguishing the type (receptor, ligand) of each partner, and using biological information on each complex, these patches are classified into 19 so-called *typed families*. Table SI-1 displays the PDB ids and the partner specifications of our dataset, while Table SI-2 presents the typed families.

Supporting Information Figure 4 The patch having the largest number of atoms. 2oza chain B possesses 395 atoms and 28 shells.



Supporting Information Figure 5 The patch having largest number of shells. 2oza chain A possesses 365 atoms and 47 shells.



Supporting Information Table 1 The 249 protein complexes used in this study.

PDB Id	Chains		Typed families	
	Partner A	Partner B	Partner A	Partner B
1mfe	HL	1	A_Carb_R	A_Carb_L
1q9q	BA	C	A_Carb_R	A_Carb_L
1q9r	BA	C	A_Carb_R	A_Carb_L
1q9t	BA	C	A_Carb_R	A_Carb_L
1s3k	HL	C	A_Carb_R	A_Carb_L
1zls	HL	X	A_Carb_R	A_Carb_L
3hns	HL	1	A_Carb_R	A_Carb_L
3hnt	HL	1	A_Carb_R	A_Carb_L
3hnv	HL	1	A_Carb_R	A_Carb_L
1a3l	HL	1	A_Chem_R	A_Chem_L
1a6w	HL	1	A_Chem_R	A_Chem_L
1c5c	HL	1	A_Chem_R	A_Chem_L
1d6v	HL	1	A_Chem_R	A_Chem_L
1flr	HL	1	A_Chem_R	A_Chem_L
1hyx	HL	1	A_Chem_R	A_Chem_L
1hyy	HL	1	A_Chem_R	A_Chem_L
1jgu	HL	1	A_Chem_R	A_Chem_L
1kn2	HL	1	A_Chem_R	A_Chem_L
1kn4	HL	1	A_Chem_R	A_Chem_L
1mex	HL	1	A_Chem_R	A_Chem_L
1n7m	LH	1	A_Chem_R	A_Chem_L
1q0y	HL	1	A_Chem_R	A_Chem_L
1q72	HL	1	A_Chem_R	A_Chem_L
1q9v	BA	1	A_Chem_R	A_Chem_L
1riu	HL	1	A_Chem_R	A_Chem_L
1wz1	HL	1	A_Chem_R	A_Chem_L
1yec	HL	1	A_Chem_R	A_Chem_L
1yef	HL	1	A_Chem_R	A_Chem_L
1yei	HL	1	A_Chem_R	A_Chem_L
1yej	HL	1	A_Chem_R	A_Chem_L
25c8	HL	1	A_Chem_R	A_Chem_L
2ajs	HL	1	A_Chem_R	A_Chem_L
2ajv	HL	1	A_Chem_R	A_Chem_L
2ajx	HL	1	A_Chem_R	A_Chem_L
2dqt	HL	1	A_Chem_R	A_Chem_L
2dqu	HL	1	A_Chem_R	A_Chem_L
2r23	BA	1	A_Chem_R	A_Chem_L
2r2b	BA	1	A_Chem_R	A_Chem_L
2r2h	BA	1	A_Chem_R	A_Chem_L
35c8	HL	1	A_Chem_R	A_Chem_L
3dv4	BA	1	A_Chem_R	A_Chem_L
3dv6	BA	1	A_Chem_R	A_Chem_L
3hzm	BA	1	A_Chem_R	A_Chem_L
3hzv	BA	1	A_Chem_R	A_Chem_L
3ls4	HL	1	A_Chem_R	A_Chem_L
3phq	BA	1	A_Chem_R	A_Chem_L
3t4y	BA	1	A_Chem_R	A_Chem_L
3t65	BA	1	A_Chem_R	A_Chem_L
3t77	BA	1	A_Chem_R	A_Chem_L

PDB Id	Chains		Typed families	
	Partner A	Partner B	Partner A	Partner B
2ok0	HL	D	A_DNA_R	A_DNA_L
1ce1	HL	P	A_Pept_R	A_Pept_L
1e4w	HL	P	A_Pept_R	A_Pept_L
1i8k	BA	C	A_Pept_R	A_Pept_L
1mvu	BA	P	A_Pept_R	A_Pept_L
1pz5	BA	C	A_Pept_R	A_Pept_L
1sm3	HL	P	A_Pept_R	A_Pept_L
1tjg	HL	P	A_Pept_R	A_Pept_L
1u8i	BA	C	A_Pept_R	A_Pept_L
2b1h	HL	P	A_Pept_R	A_Pept_L
2f5b	HL	P	A_Pept_R	A_Pept_L
2fx7	HL	P	A_Pept_R	A_Pept_L
3drq	BA	C	A_Pept_R	A_Pept_L
3eys	HL	Q	A_Pept_R	A_Pept_L
3fn0	HL	P	A_Pept_R	A_Pept_L
3g5y	BA	E	A_Pept_R	A_Pept_L
3go1	HL	P	A_Pept_R	A_Pept_L
3idg	BA	C	A_Pept_R	A_Pept_L
3ifl	HL	P	A_Pept_R	A_Pept_L
3ley	HL	P	A_Pept_R	A_Pept_L
3mlr	HL	P	A_Pept_R	A_Pept_L
3mnz	BA	P	A_Pept_R	A_Pept_L
1a2y	BA	C	A_Prot_R	A_Prot_L
1ahw	AB	C	A_Prot_R	A_Prot_L
1bvk	DE	F	A_Prot_R	A_Prot_L
1dqj	BA	C	A_Prot_R	A_Prot_L
1e6j	HL	P	A_Prot_R	A_Prot_L
1f58	HL	P	A_Prot_R	A_Prot_L
1fns	HL	A	A_Prot_R	A_Prot_L
1g7h	BA	C	A_Prot_R	A_Prot_L
1g7i	BA	C	A_Prot_R	A_Prot_L
1g7j	BA	C	A_Prot_R	A_Prot_L
1g7l	BA	C	A_Prot_R	A_Prot_L
1g7m	BA	C	A_Prot_R	A_Prot_L
1iqd	BA	C	A_Prot_R	A_Prot_L
1j1o	HL	Y	A_Prot_R	A_Prot_L
1j1p	HL	Y	A_Prot_R	A_Prot_L
1j1x	HL	Y	A_Prot_R	A_Prot_L
1jps	HL	T	A_Prot_R	A_Prot_L
1k4c	AB	C	A_Prot_R	A_Prot_L
1kiq	BA	C	A_Prot_R	A_Prot_L
1kir	BA	C	A_Prot_R	A_Prot_L
1mlc	AB	E	A_Prot_R	A_Prot_L
1nby	BA	C	A_Prot_R	A_Prot_L
1nbz	BA	C	A_Prot_R	A_Prot_L
1ndg	BA	C	A_Prot_R	A_Prot_L
1ors	BA	C	A_Prot_R	A_Prot_L
1osp	HL	O	A_Prot_R	A_Prot_L
1p2c	AB	C	A_Prot_R	A_Prot_L
1r3j	BA	C	A_Prot_R	A_Prot_L
1ua6	HL	Y	A_Prot_R	A_Prot_L

PDB Id	Chains		Typed families	
	Partner A	Partner B	Partner A	Partner B
1uac	HL	Y	A_Prot_R	A_Prot_L
1vfb	BA	C	A_Prot_R	A_Prot_L
1wej	HL	F	A_Prot_R	A_Prot_L
1yqv	HL	Y	A_Prot_R	A_Prot_L
2adf	HL	A	A_Prot_R	A_Prot_L
2dqc	HL	Y	A_Prot_R	A_Prot_L
2dqd	HL	Y	A_Prot_R	A_Prot_L
2dqe	HL	Y	A_Prot_R	A_Prot_L
2dqi	HL	Y	A_Prot_R	A_Prot_L
2dqj	HL	Y	A_Prot_R	A_Prot_L
2i25	N	L	A_Prot_R	A_Prot_L
2ih3	AB	C	A_Prot_R	A_Prot_L
2vir	AB	C	A_Prot_R	A_Prot_L
2vis	AB	C	A_Prot_R	A_Prot_L
2vxq	HL	A	A_Prot_R	A_Prot_L
2vxt	HL	I	A_Prot_R	A_Prot_L
3a67	HL	Y	A_Prot_R	A_Prot_L
3a6b	HL	Y	A_Prot_R	A_Prot_L
3a6c	HL	Y	A_Prot_R	A_Prot_L
3bae	HL	A	A_Prot_R	A_Prot_L
3d9a	HL	C	A_Prot_R	A_Prot_L
3ffd	AB	P	A_Prot_R	A_Prot_L
3h0t	BA	C	A_Prot_R	A_Prot_L
3ifn	HL	P	A_Prot_R	A_Prot_L
1acb	E	I	E_Inhi_R	E_Inhi_L
1avx	A	B	E_Inhi_R	E_Inhi_L
1ay7	A	B	E_Inhi_R	E_Inhi_L
1brs	A	D	E_Inhi_R	E_Inhi_L
1buh	A	B	E_Inhi_R	E_Inhi_L
1bvn	P	T	E_Inhi_R	E_Inhi_L
1cbw	ABC	D	E_Inhi_R	E_Inhi_L
1dfj	E	I	E_Inhi_R	E_Inhi_L
1eaw	A	B	E_Inhi_R	E_Inhi_L
1emv	A	B	E_Inhi_R	E_Inhi_L
1ezu	C	AB	E_Inhi_R	E_Inhi_L
1f34	A	B	E_Inhi_R	E_Inhi_L
1fle	E	I	E_Inhi_R	E_Inhi_L
1gl1	A	I	E_Inhi_R	E_Inhi_L
1gxd	A	C	E_Inhi_R	E_Inhi_L
1hia	AB	I	E_Inhi_R	E_Inhi_L
1jiw	P	I	E_Inhi_R	E_Inhi_L
1jtg	B	A	E_Inhi_R	E_Inhi_L
1mah	A	F	E_Inhi_R	E_Inhi_L
1nb5	AP	I	E_Inhi_R	E_Inhi_L
1oph	A	B	E_Inhi_R	E_Inhi_L
1ppe	E	I	E_Inhi_R	E_Inhi_L
1pxv	A	C	E_Inhi_R	E_Inhi_L
1r0r	E	I	E_Inhi_R	E_Inhi_L
1uug	A	B	E_Inhi_R	E_Inhi_L
1yvb	A	I	E_Inhi_R	E_Inhi_L
1zli	A	B	E_Inhi_R	E_Inhi_L

PDB Id	Chains		Typed families	
	Partner A	Partner B	Partner A	Partner B
2abz	B	E	E_Inhi_R	E_Inhi_L
2b42	A	B	E_Inhi_R	E_Inhi_L
2j0t	A	D	E_Inhi_R	E_Inhi_L
2o3b	A	B	E_Inhi_R	E_Inhi_L
2oul	A	B	E_Inhi_R	E_Inhi_L
2ptc	E	I	E_Inhi_R	E_Inhi_L
2sic	E	I	E_Inhi_R	E_Inhi_L
2sni	E	I	E_Inhi_R	E_Inhi_L
2tgp	Z	I	E_Inhi_R	E_Inhi_L
2uuy	A	B	E_Inhi_R	E_Inhi_L
2wpt	A	B	E_Inhi_R	E_Inhi_L
3sgb	E	I	E_Inhi_R	E_Inhi_L
4cpa	A	I	E_Inhi_R	E_Inhi_L
1gla	G	F	E_Regu_R	E_Regu_L
1ijk	A	BC	E_Regu_R	E_Regu_L
1jmo	A	HL	E_Regu_L	E_Regu_R
1jwh	CD	A	E_Regu_L	E_Regu_R
1m10	A	B	E_Regu_R	E_Regu_L
1nw9	B	A	E_Regu_R	E_Regu_L
1oc0	A	B	E_Regu_R	E_Regu_L
1r6q	A	C	E_Regu_R	E_Regu_L
1us7	A	B	E_Regu_R	E_Regu_L
1wdw	BD	A	E_Regu_R	E_Regu_L
2oor	AB	C	E_Regu_R	E_Regu_L
1e6e	A	B	E_Subs_R	E_Subs_L
1ewy	A	C	E_Subs_R	E_Subs_L
1f6m	A	C	E_Subs_R	E_Subs_L
1kkl	ABC	H	E_Subs_R	E_Subs_L
1zm4	B	A	E_Subs_R	E_Subs_L
2a9k	B	A	E_Subs_R	E_Subs_L
2mta	HL	A	E_Subs_R	E_Subs_L
2oob	A	B	E_Subs_R	E_Subs_L
2pcb	A	B	E_Subs_R	E_Subs_L
2pcc	A	B	E_Subs_R	E_Subs_L
1a2k	C	AB	OG	OG
1e96	A	B	OG	OG
1fqj	A	B	OG	OG
1grn	A	B	OG	OG
1he8	B	A	OG	OG
1i2m	A	B	OG	OG
1i4d	D	AB	OG	OG
1ibr	A	B	OG	OG
1j2j	A	B	OG	OG
1k5d	AB	C	OG	OG
1lfd	B	A	OG	OG
1nvu	Q	S	OG	OG
1nvu	R	S	OG	OG
1wq1	R	G	OG	OG
1z0k	A	B	OG	OG
2fju	B	A	OG	OG
3cph	G	A	OG	OG

PDB Id	Chains		Typed families	
	Partner A	Partner B	Partner A	Partner B
1e4k	AB	C	OR	OR
1eer	A	BC	OR	OR
1hcf	AB	X	OR	OR
1kac	A	B	OR	OR
1ktz	A	B	OR	OR
1pvh	A	B	OR	OR
1rv6	VW	X	OR	OR
1t6b	X	Y	OR	OR
1xu1	ABD	T	OR	OR
2ajf	A	E	OR	OR
2hle	A	B	OR	OR
2i9b	E	A	OR	OR
2nyz	AB	D	OR	OR
1ak4	A	D	OX	OX
1akj	AB	DE	OX	OX
1atn	A	D	OX	OX
1avz	B	C	OX	OX
1b6c	A	B	OX	OX
1de4	AB	CF	OX	OX
1efn	B	A	OX	OX
1fc2	C	D	OX	OX
1ffw	A	B	OX	OX
1gcq	B	C	OX	OX
1gpw	A	B	OX	OX
1h1v	A	G	OX	OX
1h9d	A	B	OX	OX
1ib1	AB	E	OX	OX
1klu	AB	D	OX	OX
1kxp	A	D	OX	OX
1mq8	A	B	OX	OX
1qa9	A	B	OX	OX
1rlb	ABCD	E	OX	OX
1s1q	A	B	OX	OX
1xd3	A	B	OX	OX
1xqs	A	C	OX	OX
1zhi	A	B	OX	OX
2aq3	A	B	OX	OX
2b4j	AB	C	OX	OX
2btf	A	P	OX	OX
2c0l	A	B	OX	OX
2gox	A	B	OX	OX
2hqs	A	H	OX	OX
2hrk	A	B	OX	OX
2oza	B	A	OX	OX
2vdb	A	B	OX	OX
3bp8	AB	C	OX	OX
3bzd	A	B	OX	OX

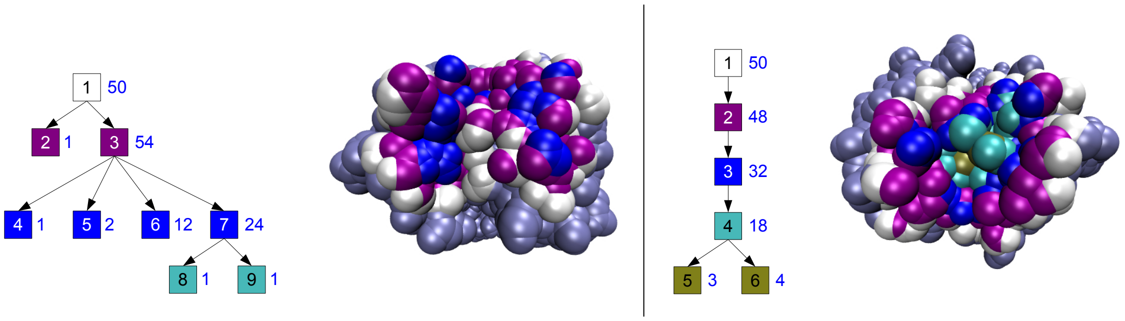
Supporting Information Table 2 The typed family classification of the 498 patches from our database. Stars indicate the typed families used for the identification and clustering experiments. The OG, OR and OX classes are from [17].

Family of complex	Sub-Family of complex	Partner Type	Class identifier	#patches
(A) Antibody	(Carb) Carbohydrate	(R) Receptor	A_Carb_R *	9
		(L) Ligand	A_Carb_L *	9
	(Chem) Chemical	(R) Receptor	A_Chem_R *	40
		(L) Ligand	A_Chem_L *	40
	(DNA) DNA	(R) Receptor	A_DNA_R	1
		(L) Ligand	A_DNA_L	1
	(Pept) Peptide	(R) Receptor	A_Pept_R *	21
		(L) Ligand	A_Pept_L *	21
	(Prot) Protein	(R) Receptor	A_Prot_R *	53
		(L) Ligand	A_Prot_L *	53
(E) Enzyme	(Inhi) Inhibitor	(R) Receptor	E_Inhi_R *	40
		(L) Ligand	E_Inhi_L *	40
	(Regu) Regulator	(R) Receptor	E_Regu_R *	11
		(L) Ligand	E_Regu_L *	11
	(Subs) Substrat	(R) Receptor	E_Subs_R *	10
		(L) Ligand	E_Subs_L *	10
(O) Other	(G) G-prot. containing	non-available	OG	34
	(R) Recept. containing	non-available	OR	26
	(X) Misc.	non-available	OX	68

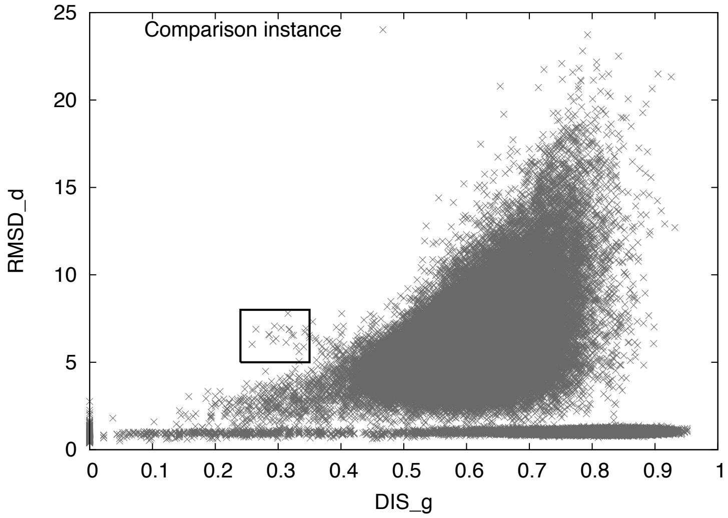
6.5 Dissecting the Morphology of Patches

6.6 Comparing Patches

Supporting Information Figure 6 Two patches having similar topologies but different geometries. Left: patch of 1dqj chains BA. Right: patch of 1jps chains HL. Their topological dissimilarity DIS_t is about 0.18 (due to shells 1-3-7 from 1dqj_BA that match shells 1-2-3 from 1jps_HL), but their geometric dissimilarity DIS_g is about 0.68.



Supporting Information Figure 7 Geometric dissimilarity ($\mathbf{DIS}_g$) versus RMSD of internal distances ($RMSD_d$) computed at $\epsilon = 2$Å. The $RMSD_d$ globally increases with the geometric dissimilarity, and so does the variance of the $RMSD_d$ for a fixed DIS_g. The black box singles out instances with low geometric dissimilarity but high $RMSD_d$ values.



6.7 Identifying Patches and Specificity Analysis

Supporting Information Table 3 Correct identification rates per typed family. The typed family of a patch is identified by the typed 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 family the number (resp percentage) of correctly identified binding patches.

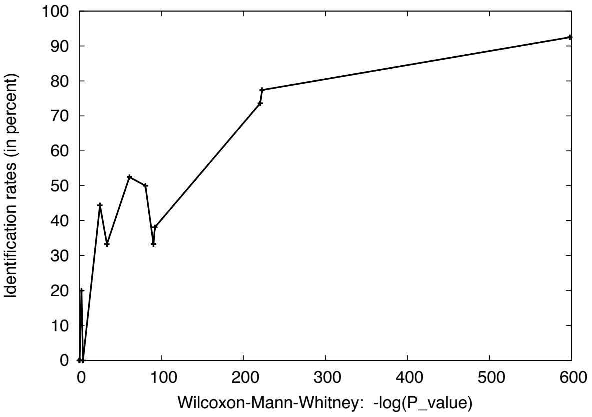
Typed family	Correctly identified patches	percentage
A_Carb_R	4/9	44.4%
A_Carb_L	3/9	33.3%
A_Chem_R	37/40	92.5%
A_Chem_L	34/40	85.0%
A_Pept_R	7/21	33.3%
A_Pept_L	8/21	38.1%
A_Prot_R	39/53	73.6%
A_Prot_L	41/53	77.4%
E_Inhi_R	20/40	50%
E_Inhi_L	21/40	52.5%
E_Regu_R	0/11	0%
E_Regu_L	0/11	0%
E_Subs_R	2/10	20%
E_Subs_L	0/10	0%

The low identification rates can be explained by inconsistencies in the original classification. Such an inconsistency is presented on Fig. SI-6, where both 1jps chains HL and 1dqj chains BA come from the AA_Prot_R typed family, but only have about 32% of geometrically common atoms.

Supporting Information Table 4 p-values for the Wilcoxon-Mann-Whitney rank-sum test for similarity scores obtained by TED$_g$ with ϵ=2Å. The null hypothesis (identical distributions) is rejected for all the immunoglobulin families, while in the case of the enzymes, it is only rejected for the E_Inhi family. Note that these probabilities closely follow the identification rates observed in table SI-3.

Family (=P)	$s(P,P)$ vs $s(P,\overline{P})$	$s(P,P)$ vs $s(P,P^C)$
A_Carb_R	1.96e-10	1.04e-11
A_Carb_L	2.91e-8	2.20e-15
A_Chem_R	1.98e-266	1.74e-260
A_Chem_L	8.73e-225	0
A_Pept_R	6.12e-29	4.94e-40
A_Pept_L	9.98e-32	8.41e-41
A_Prot_R	8.82e-5	1.45e-96
A_Prot_L	1.07e-35	1.50e-97
E_Inhi_R	0.05	9.10e-36
E_Inhi_L	9.41e-19	2.53e-27
E_Regu_R	0.22	0.96
E_Regu_L	0.94	0.60
E_Subs_R	0.06	0.06
E_Subs_L	0.75	0.01

Supporting Information Figure 8 The quality of the identification is related to the consistency of the families. For each typed family P, the correct identification rate is plotted as a function of $-log(P_{value})$, P_{value} being the p-value of the Wilcoxon-Mann-Whitney rank-sum test between $s(P,P)$ and $s(P,P^c)$. Very low P_{values} (consistent families) relate to high identification rates.



6.8 Predicting Binding Affinities

Supporting Information Table 5 The Internal Path Length yields the best correlation against the binding affinity ($-\ln K_d$). The values were computed for the 144 complexes of the affinity benchmark. Note that MIC p-values are available only for coefficient larger than 0.275, see `http://www.exploredata.net/Downloads/P-Value-Tables`.

	Pearson		Spearman		Maximal Information	
Parameter	C_{Pea}	p-value	C_{Spe}	p-value	C_{MIC}	p-value
IPL	0.31	$1.3e{-}4$	0.43	$7.6e^{-8}$	0.35	$7.6e^{-4}$
#Atoms	0.27	$1.2e{-}3$	0.37	$4.7e^{-6}$	0.24	
Depth	0.29	$4.8e{-}4$	0.35	$1.5e^{-5}$	0.26	
ΔASA	0.22	$8.9e{-}3$	0.33	$6.6e^{-5}$	0.25	
Firedock score	-0.17	$4.2e{-}2$	0.20	$1.8e^{-2}$	0.23	
I_RMSD	-0.11	$2.0e{-}1$	0.17	$4.3e^{-2}$	0.24	
#Shells	0.092	$2.7e{-}1$	-0.16	$5.4e^{-2}$	0.16	
DIS_g	0.16	$5.8e{-}2$	-0.14	$8.5e^{-2}$	0.24	
Assymetry	0.045	$5.9e{-}1$	-0.094	$2.6e^{-1}$	0.19	
DIS_t	0.029	$7.2e{-}1$	-0.089	$2.9e^{-1}$	0.20	

Supporting Information Table 6 Spearman's correlation coefficient as a function of the flexibility observed upon binding. Spearman coefficient between the binding affinity $-\ln K_d$ and ΔASA, #Atoms, Depth and IPL, for the three flexibility classes introduced in [17]: I-RMSD $<$ 1 (rigid interfaces, 71 cases), I-RMSD $\in$ [1 , 1.5 [(semi-rigid interfaces, 38 cases), and I-RMSD $\geq$ 1.5 (flexible interfaces, 35 cases).

	ΔASA		#Atoms		Depth		IPL	
I-RMSD (Å)	C_{Spe}	p-value	C_{Spe}	p-value	C_{Spe}	p-value	C_{Spe}	p-value
$<$ 1 Å	0.52	$3.5e^{-6}$	0.58	$1.4e^{-7}$	0.54	$9.0e^{-7}$	0.59	$5.9e^{-8}$
in [1Å,1.5Å[	0.18	$2.7e^{-1}$	0.11	$5.0e^{-1}$	0.054	$7.5e^{-1}$	0.23	$1.7e^{-1}$
$\geq$ 1.5Å	0.26	$1.2e^{-1}$	0.34	$4.7e^{-2}$	0.34	$4.2e^{-2}$	0.41	$1.5e^{-2}$

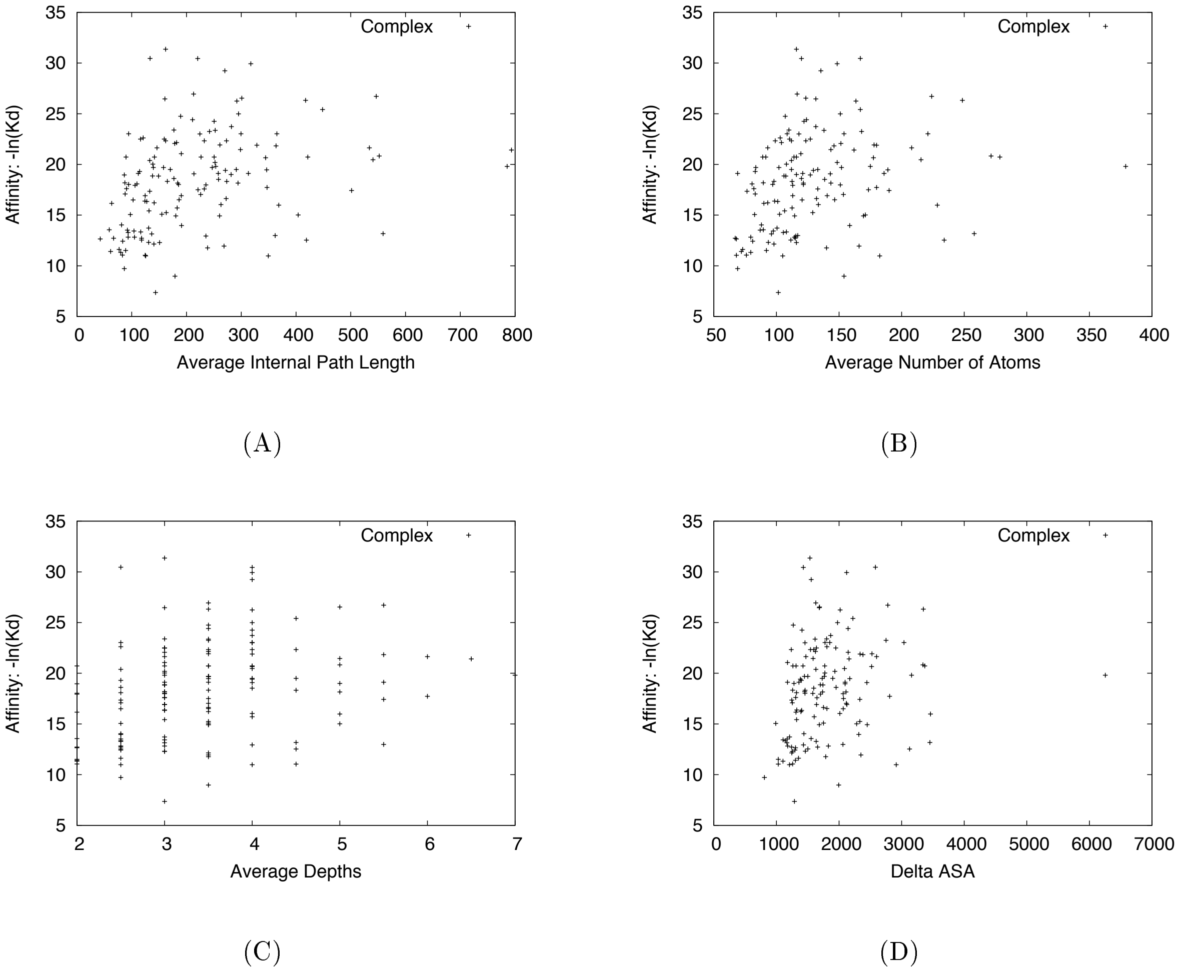
Supporting Information Table 7 Maximal Information Coefficient as a function of the flexibility observed upon binding. MIC between the binding affinity $-\ln K_d$ and ΔASA, #Atoms, Depth and IPL, for the three flexibility classes introduced in [17]: I-RMSD $<$ 1 (rigid interfaces, 71 cases), I-RMSD $\in$ [1 , 1.5 [(semi-rigid interfaces, 38 cases), and I-RMSD $\geq$ 1.5 (flexible interfaces, 35 cases).

	ΔASA		#Atoms		Depth		IPL	
I-RMSD (Å)	C_{MIC}	p-value	C_{MIC}	p-value	C_{MIC}	p-value	C_{MIC}	p-value
$<$ 1 Å	0.39	$1.5e^{-2}$	0.43	$3.6e^{-3}$	0.43	$3.2e^{-3}$	0.48	$4.4e^{-4}$
in [1Å,1.5Å[	0.60	$3.6e^{-4}$	0.26	_	0.28	_	0.34	_
$\geq$ 1.5Å	0.22	_	0.36	_	0.33	_	0.31	_

Supporting Information Table 8 Assessment of the geometric (dis-)similarity of pairs of closely related proteins forming complexes with a very different K_d—data from [17, Table 2].

cognate	K_d	non cognate	K_d	DIS_g recep	DIS_g lig
1brs__A-D	$5.0e^{-14}$	1ay7__A-B	$1.0e^{-10}$	0.52	0.46
1emv__A-B	$2.4e^{-14}$	2wpt__A-B	$1.5e^{-8}$	0.36	0.38
2ptc__E-I	$6.0e^{-14}$	1cbw__ABC-D	$9.0e^{-9}$	0.47	0.31
2ptc__E-I	$6.0e^{-14}$	2tgp__Z-I	$2.3e^{-6}$	0.20	0.24
2pcc__A-B	$1.6e^{-6}$	2pcb__A-B	$1.0e^{-5}$	0.59	0.49
2vir__AB-C	$1.0e^{-9}$	2vis__AB-C	$4.0e^{-6}$	0.35	0.23
1p2c__AB-C	$1.0e^{-10}$	1mlc__AB-E	$7.0e^{-8}$	0.51	0.45
1efn__B-A	$3.8e^{-8}$	1avz__B-C	$1.6e^{-5}$	0.45	0.48
3bzd__A-B	$9.6e^{-8}$	2aq3__A-B	$1.2e^{-5}$	0.55	0.51

Supporting Information Figure 9 Scatter plots of binding affinity against various structural parameters (A) Affinity versus Internal Path Length ($C_{\mathrm{Pea}} = 0.31$ and $C_{\mathrm{Spe}} = 0.43$). (B) Affinity versus number of atoms ($C_{\mathrm{Pea}} = 0.27$ and $C_{\mathrm{Spe}} = 0.37$) (C) Affinity versus depth ($C_{\mathrm{Pea}} = 0.29$ and $C_{\mathrm{Spe}} = 0.35$) (D) Affinity versus ΔASA ($C_{\mathrm{Pea}} = 0.22$ and $C_{\mathrm{Spe}} = 0.33$). Note that A, B and C use the average value of the considered parameter across the two partner patches of each complex.



(A)

(B)

(C)

(D)

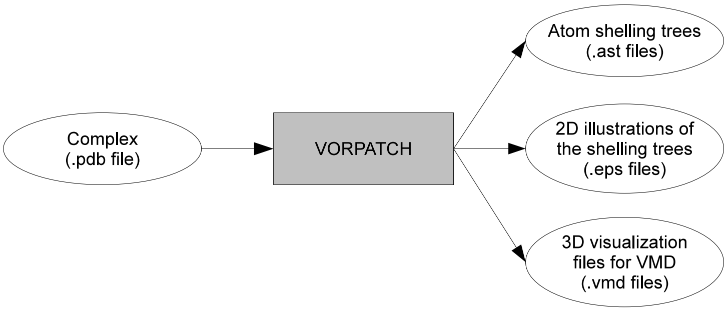
6.9 Software: application and file formats

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/.

6.9.1 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 6.1.3. The atom shelling trees are recorded in the custom file format described in the supplemental Fig. 2. 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 1 Overview of the patch generation with COMPATCH.

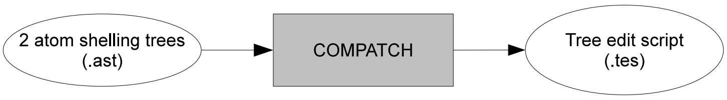


Supplemental Figure 2 The atom shelling tree (.ast) file format. An atom shelling tree is described by the list of its nodes (or shells). Each node is first described by a header line containing three integers: the node ID (starting from 1), the number of atom of the corresponding shell, and the node's 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
# ...
```

6.9.2 **Program** COMPATCH

Supplemental Figure 3 Overview of the patch comparison with COMPATCH**.**



Given two patches (.ast file format), COMPATCH use the tree-edit-distance based methods presented in section 6.2.1 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. 4. 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 4 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)
```

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399