HAL
open science

On the Characterization and Selection of Diverse Conformational Ensembles, with Applications to Flexible Docking

Sebastien Loriot, Sushant Sachdeva, Karine Bastard, Chantal Prevost, Frédéric Cazals

To cite this version:

Sebastien Loriot, Sushant Sachdeva, Karine Bastard, Chantal Prevost, Frédéric Cazals. On the Characterization and Selection of Diverse Conformational Ensembles, with Applications to Flexible Docking. [Research Report] RR-6503, INRIA. 2009. inria-00252046v3

HAL Id: inria-00252046

https://inria.hal.science/inria-00252046v3

Submitted on 6 Apr 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On the Characterization and Selection of Diverse Conformational Ensembles

Sébastien Loriot — Sushant Sachdeva — Karine Bastard — Chantal Prévost — Frédéric Cazals

N° 6503 — version 2

version initiale February 2008 — version révisée April 2008

Thème BIO

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6503--FR+ENG

On the Characterization and Selection of Diverse Conformational Ensembles

Sébastien Loriot [*], Sushant Sachdeva [†], Karine Bastard [‡], Chantal Prévost [§], Frédéric Cazals [¶]

Thème BIO — Systèmes biologiques
Projet ABS

Rapport de recherche n° 6503 — version 2 — version initiale February 2008 — version révisée April 2008 — 39 pages

Abstract: To address challenging flexible docking problems, a number of docking algorithms pre-generate large collections of candidate conformers. To remove the redundancy from such ensembles, a central problem in this context is to report a selection of conformers maximizing some geometric diversity criterion. We make three contributions to this problem.

First, we resort to geometric optimization so as to report selections maximizing the molecular volume or molecular surface area (MSA) of the selection. Greedy strategies are developed, together with approximation bounds.

Second, to assess the efficacy of our algorithms, we investigate two conformer ensembles corresponding to a flexible loop of four protein complexes. By focusing on the MSA of the selection, we show that our strategy matches the MSA of standard selection methods, but resorting to a number of conformers between one and two orders of magnitude smaller. This observation is qualitatively explained using the Betti numbers of the union of balls of the selection.

Finally, we replace the conformer selection problem in the context of multiple-copy flexible docking. On the afore-mentioned systems, we show that using the loops selected by our strategy can improve the result of the docking process.

Key-words: Flexibility, conformer selection, flexible docking, geometric optimization, Van der Waals models.

[*] INRIA Sophia-Antipolis, Algorithms-Biology-Structure, France and Université de Bourgogne IMB, France; Sebastien.Loriot@sophia.inria.fr
[†] IIT Bombay, India; sachdevas@cse.iitb.ac.in
[‡] Biotechnologie-Biocatalyse-Biorégulation, Université de Nantes - CNRS, France; Karine.Bastard@univ-nantes.fr
[§] Institut de Biologie Physico-Chimique, Paris, France; Chantal.Prevost@ibpc.fr
[¶] INRIA Sophia-Antipolis, Algorithms-Biology-Structure, France; Frederic.Cazals@inria.fr

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Sur la caractérisation et la sélection d'ensembles variables de conformères

Résumé : Pour traiter les problèmes liés aux méthodes d'amarrage de protéines avec parties flexibles, nombre d'algorithmes utilisent un ensemble de grande taille de conformations pré-calculées. Afin de réduire la redondance au sein d'un tel ensemble, une question centrale est la suivante : Sélectionner un sous-ensemble (de petite taille) de conformations qui maximisent un critère géométrique de diversité. Dans cette optique, nous présentons trois contributions.

Premièrement, nous posons le problème précédent comme un problème d'optimisation géométrique, de façon à sélectionner des conformations qui maximisent le volume moléculaire ou l'aire de la surface moléculaire de la sélection. Des stratégies gloutonnes sont développées pour résoudre ces problèmes, et les bornes d'approximation afférentes sont prouvées.

Deuxièmement, pour montrer l'efficacité de nos algorithmes, nous effectuons des tests sur quatre complexes protéiques présentant une boucle flexible, deux ensembles de conformations étant générés pour chaque boucle. Pour le critère basé sur l'aire de la surface moléculaire, nous montrons que nos sélections présentent des aires semblables à celles de sélections réalisées par des méthodes standards, mais en utilisant un nombre de conformations entre un et deux ordres de grandeur plus faible. Cette observation est qualitativement expliquée en faisant appel aux nombres de Betti de l'union des boules de la sélection.

Enfin, pour le problème de l'amarrage de protéines avec parties flexibles, nous montrons sur les mêmes modèles que notre méthode permet d'améliorer les résultats d'une telle procédure.

Mots-clés : Flexibilité, selection de conformères, docking flexible, optimisation géométrique, modèles de Van der Waals.

1 Introduction

1.1 On the Importance of Diverse Conformational Ensembles

Ensembles in molecular modeling. Protein-protein interactions are paramount to all biological processes, but their prediction from unbound geometries faces major difficulties, as evidenced in the CAPRI experiment, by the low number of *medium* and *high* predictions carried out on flexible systems—as opposed to *incorrect* and *acceptable* ones [JW07]. Since proteins are intrinsically flexible, they continuously undergo conformational changes over time, or in an equivalent way, they exist at a given time as an ensemble of conformations in equilibrium. During their exploration of the conformational space, they preferably occupy regions which are characterized by low free energies. For proteins of moderate size undergoing small amplitude movements occurring in time scales of tens of nanoseconds, conformational changes can be investigated using molecular dynamics, namely by numerically integrating Newton's equations of motion. For more complex cases, where flexibility applies to large parts of the protein backbone or where the amplitude of the movement is important, discrete ensembles of conformations known as *conformers* can be pre-generated and considered simultaneously. This representation is particularly appropriate when dealing with macromolecular docking. In the case of association, one indeed wishes to predict the best possible bound geometry of two flexible objects, which subsumes exploring the relative position and orientation of the partners, but also their conformational space so as to pack the interface. In the Monod-Wyman-Changeux interpretation [MWC65], the unbound proteins are considered as two collections of conformers in thermodynamic equilibrium. When the partners bind, the equilibrium is shifted toward the structure observed in the complex. Implementing this strategy may be done at the global (i.e. protein) scale [GLN04], local (i.e. side chain) scale [CSD03], or intermediate (i.e. loops or domains) scale [BPZ06].

Generating and selecting conformers: energy versus geometry. Representing flexibility through an ensemble of conformers is computationally feasible only if the size of this ensemble is not too important. It is therefore essential for this small number of conformers to be as representative as possible of the conformational space available to the flexible molecule or molecular fragment. More generally, conformers being of interest for several applications, which criteria (geometric or energetic) should one use to generate and/or select them? From a statistical viewpoint, energy should be the criterion of choice for generating ensembles representative of the thermodynamic equilibrium between conformations. However, this criteria is generally not tractable for several reasons.

First and foremost, the exhaustive exploration of the conformational space of large systems or of systems with large amplitude deformations is not possible. To keep calculations tractable, methods undertaking this task favor geometric calculations, and defer energy calculations to later stages [CSRdA+05, DYM+07]. That is, a two-stage strategy is in order: first, a geometry-based exploration is favored; next, more subtle energetic considerations are taken into account. The strategy developed in this work falls in the first realm and is meant to be complemented by more delicate molecular dynamics or Monte Carlo based calculations. Second, when conformers are used to model a region of a protein, the energy associated to each conformer varies with its environment. In the case of docking for example, the energy of each copy depends upon its interactions with the partner of association (direct electrostatic or Van der Waals interactions, modification of the dielectric environment, desolvation energy). Therefore, weighting a conformer as if it were alone does not, in general, precisely account for its probability of occurrence. Third, it may happen that the energy landscape associated to a flexible protein is rather flat, with very small energy barriers between the conformers. In contrast to flipping between well separated conformers, the protein flexible fragment can largely explore the available space. In this case, it is important to be able to sample exhaustively the space available to the flexible fragment.

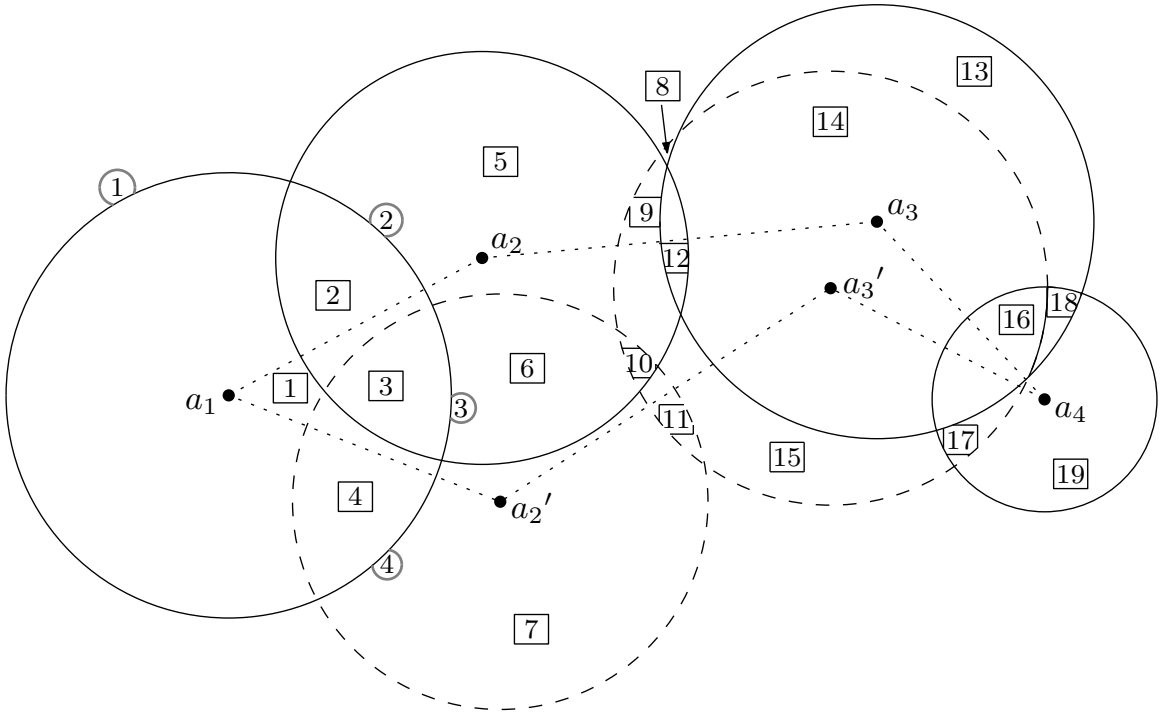
In passing, we may also notice that the generation of diverse ensembles is a strategy of choice to simulate complex processes. For example, diverse ensembles generated using a repulsive umbrella potential have recently been used to investigate domain swapping [MSLW08].

1.2 Contributions and Paper Overview

Conformers: atomic and coarse models. Consider a collection $\mathcal{C} = \{C_1, \ldots, C_n\}$ of n conformers (rotamers, protein loops, whole protein), each represented by a collection of balls, each ball being bounded by a sphere. This model is rather general: the balls in Van der Waals models represent atoms; the balls in coarse models represent residues. In this study, we use atomic and coarse protein models.

Problem addressed. As just argued, sampling the conformational space available is an important requirement. We actually wish to solve the following problem:

Figure 1 Example of $2D$ conformers, each consisting of four balls—first and fourth balls are common. The (two dimensional) volume occupied by the two conformers is decomposed into 19 cells (boxed numerals). The circled numerals feature the surface arrangement of the ball centered at a_1, based on intersections with neighboring balls.



Given a pre-computed collection of n conformers and an integer $s < n$, report a *selection* of s conformers maximizing some geometric diversity criterion.

To specify the type of geometric criterion we have in mind, observe that the union of the balls of the conformers in the selection defines a volume, whose partition by the spheres bounding the balls is called a *volumetric arrangement* (also called *volumetric decomposition*). Similarly, the decomposition of each sphere by the intersection circles with other spheres defines a *surface arrangement* (also called *surface decomposition*). See Figs. 1 and 2 for a $2D$ illustration. Using these arrangements, we investigate two geometric optimization problems whose output is the selection. These problems aim at maximizing the *spatial occupancy* of the selection, in several guises. In particular, we wish to report the s conformers maximizing (i) the volume occupied by these conformers, (ii) or the molecular surface area (MSA) of the union of the conformers, that is the surface area of the boundary of the union of balls defining these conformers. See Fig. 4 for an illustration.

As an illustration, consider Fig. 3(a), which features 40 conformers of the flexible loop of a complex. A number of these loops are obviously redundant, and one would like to trim this set to select a diverse subset. Such a selection, generated by one of our algorithms, is presented on Fig. 3(b) and Fig. 4.

Paper overview. Two conformer selection problems phrased as geometric optimization problems are presented in section 2, together with a general strategy to solve them, the *greedy strategy*. In section 3, we develop algorithm `Greedy`, which consists of reporting a selection maximizing the MSA of the union of the conformers, and present the protein-protein complexes used for the validation. A geometric and topological assessment of the diversity is presented in section 4, while an assessment of the quality of the conformers selected for flexible protein docking is presented in section 5. These assessments are conducted by comparing our algorithm, `Greedy`, to a contender named `HClust` based on a hierarchical clustering strategy. Upon concluding in section 6, we provide the proofs of the theorems presented in the main text in appendices 7 and 8, and further discuss the conformer generation methods used in appendix 9.

Figure 2 The boundary of the union of balls of the two conformers of Fig. 1, respectively in solid and dashed lines.

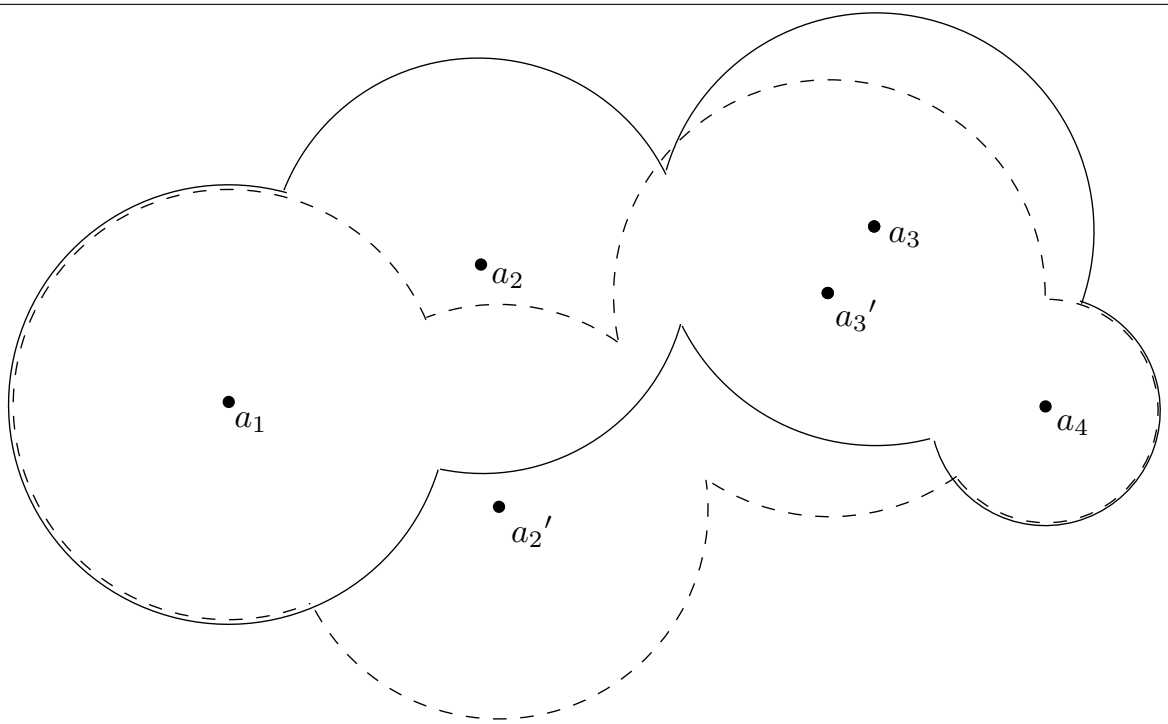


Figure 3 Selecting diverse conformational ensembles from a pool of conformers of a flexible loop—PDB code `1BTH`. From left to right: (a) backbone in cartoon mode, with 40 conformers from the pool (b) backbone together with 10 conformers selected by our algorithm—called `Greedy` (c) backbone with 10 conformers selected by a standard hierarchical clustering algorithm—called `HClust`.

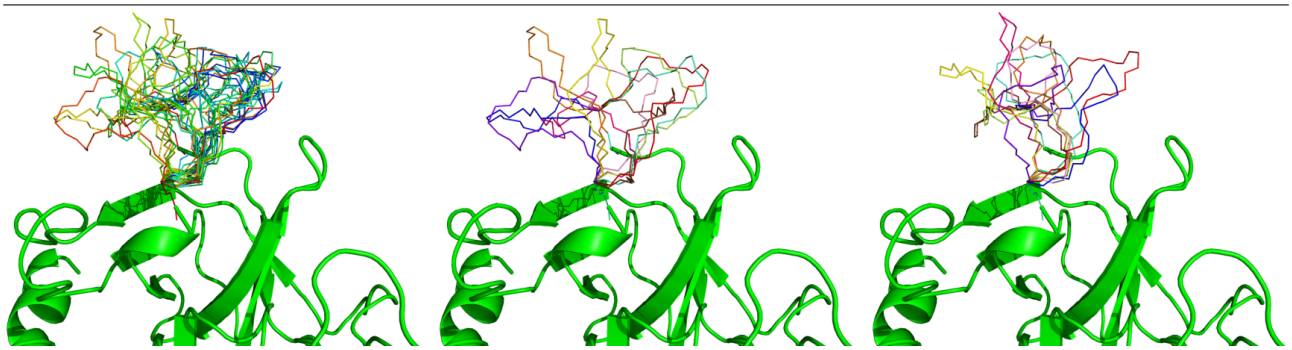
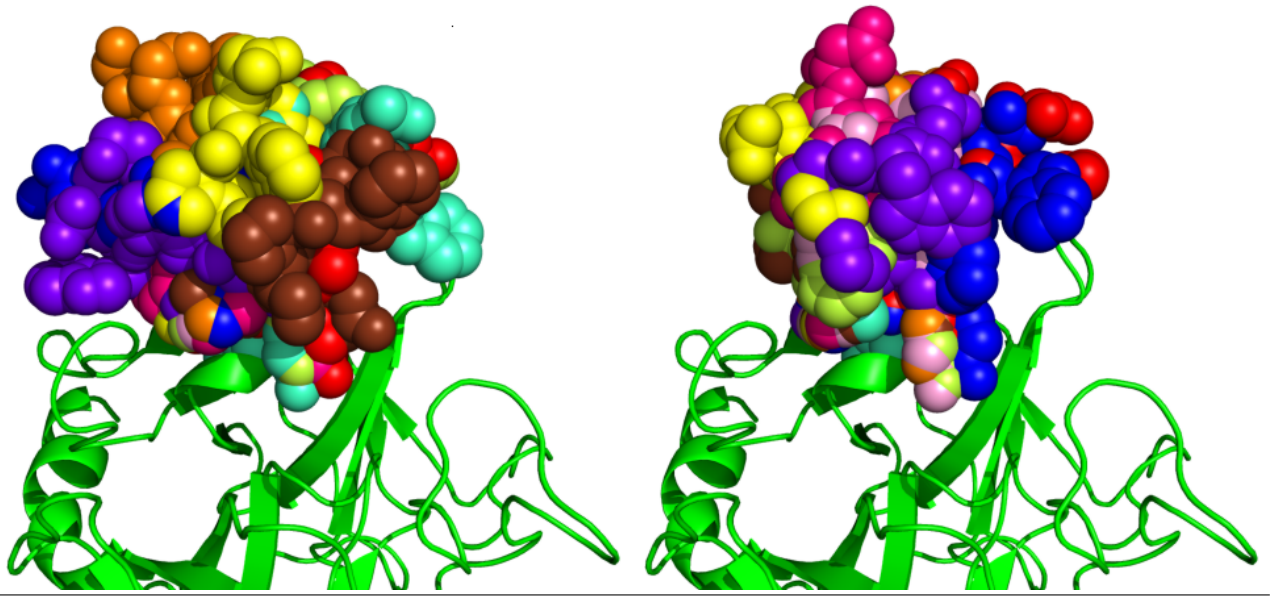


Figure 4 Follow-up to Fig. 3: Van der Waals representations of the 10 conformers presented on Fig. 3(b) and Fig. 3(c). Notice that conformers of the former set are well separated, while those of the latter are cluttered.



2 Selecting Conformers: the Combinatorial Viewpoint

2.1 Arrangements of Balls and Spheres: Volume and Surface Decompositions

The spheres bounding the balls of a collection of conformers induce two decompositions: a decomposition of the volume occupied by the balls; and a decomposition of each sphere into spherical patches. More precisely, consider the three-dimensional domain spanned by the conformers that is the union of their defining balls. The decomposition of this volume induced by the spheres is called a *volumetric arrangement* (or *volumetric decomposition*). This arrangement consists of a collection of cells $\mathcal{A} = \{A_i\}$ such that the interior of each cell is connected. Each such cell is bounded by $2D$ cells, called surface patches, found on the spheres bounding the balls. On a given sphere, these patches are induced by the intersection circles with neighboring spheres. The collection $\mathcal{P} = \{P_i\}$ of all such patches defines a *surface arrangement* (or *surface decomposition*). See Fig. 1 for an illustration.

2.2 Optimization Problems

Problems statements. We shall be concerned with two classes of combinatorial optimization problems arising from geometric representations of molecular shapes. To state these problems from a combinatorial viewpoint (see section 2.3 for the connexion with conformers), assume we are given a base set $\mathcal{U} = \{U_i\}_{i=1,\dots,m}$ of interior disjoint *cells* (think cells of the volume or surface arrangement), and a collection of *sets* $\mathcal{C} = \{C_i\}_{i=1,\dots,n}$ called the *pool* (think conformers), where each set is a union of cells. For a subset $\mathcal{S} \subset \mathcal{C}$, denote $\cup_{\mathcal{S}} C_j$ the union of the sets in $\mathcal{S}$. Cells and sets shall be subsets of $\mathbb{R}^3$, so that the inclusion of a cell U_i in a set C_j is naturally defined.

For the first class of problems, assume we are given a weight function w, i.e. a real valued function defined over the cells. Denote $\binom{\mathcal{C}}{s}$ the subsets of $\mathcal{C}$ of size s. We define:

Problem 1. *Given a weight function w, find a subset $\hat{\mathcal{S}}$ of $\mathcal{C}$ of size s, called the* selection*, such that:*

$$\hat{\mathcal{S}} = \arg\max_{\mathcal{S} \in \binom{\mathcal{C}}{s}} w(\mathcal{S}), \ \textit{with } w(\mathcal{S}) = \sum_{U_i \subset \cup_{\mathcal{S}} C_j} w(U_i). \tag{1}$$

For the second class of problems, assume the weight function depends not only on the cells of the decomposition, but also on the selection $\mathcal{S}$, which we denote $w_{\mathcal{S}}(U_i)$. We wish to solve:

Problem 2. *Given a weight function $w_{\mathcal{S}}$, find a subset $\hat{\mathcal{S}}$ of $\mathcal{C}$ of size s, called the* selection*, such that:*

$$\hat{\mathcal{S}} = \arg\max_{\mathcal{S}\in\binom{\mathcal{C}}{s}} w(\mathcal{S}), \text{ with } w(\mathcal{S}) = \sum_{U_i \subset \cup_{\mathcal{S}} C_j} w_{\mathcal{S}}(U_i). \tag{2}$$

Complexity issues. Our problems are intimately related to *max-k cover*. Given a set $\mathcal{U}$ of n points, and a collection $\mathcal{C}$ of subsets of $\mathcal{U}$, max-k cover is the problem of selecting k subsets from $\mathcal{C}$ such that the union contains as many points from $\mathcal{U}$ as possible [GJ79, Fei98]. (There is some confusion in the literature, as this problem is called set cover in [FG89]. In fact, the partial set cover problem consists of picking the minimum number of sets in $\mathcal{C}$ so as to contain at least k elements from $\mathcal{U}$.)

If the weight function w assigns a unit weight to all cells, then Problem 1 reduces to max-k cover. Since this is a **NP**-Complete problem, we cannot expect to have an exact algorithm for our problem that works in time polynomial in both $|\mathcal{C}|$ and s. On the other hand, for a fixed s, the search space which consists of all subsets of $\mathcal{C}$ of size s has size $\mathcal{O}(|\mathcal{C}|^s)$. Hence, for a fixed s the problem is in **P**. However, even for a modest s, the brute force method is too costly to be used in practice.

Section 2.4 presents an approximate strategy whose time complexity does not grow exponentially with s.

2.3 Instantiations to Conformer Selection

Problem 1 from volumetric decomposition. Consider the base set $\mathcal{A}$ whose cells are those of the $3D$ arrangement. In Eq. (1), let w be some general function defined on the cells of the volumetric decomposition, for example the standard Euclidean volume. For conformer selection, optimizing the volume of a selection is a direct way to aim for a good spatial diversity, since overlaps between conformers are minimized.

Problem 2 from surface decomposition. Consider the base set $\mathcal{P} = \{P_i\}$ whose cells are those of the $2D$ arrangements. Special cells of this arrangement are those which are exposed, i.e. contribute to the boundary of the union of balls. Focusing on these patches yields an instantiation of Problem 2, the dependence upon the selection $\mathcal{S}$ consisting of discarding the patches which are not exposed with respect to the selection. For example, $w_{\mathcal{S}}(P_i)$ = surface area of patch P_i iff P_i is found on the boundary of the union $\cup_{\mathcal{S}} C_j$, and 0 otherwise. Practically, we shall be dealing with atomic and coarse models. By molecular surface, we refer to the Van der Waals surface for the former, and to the boundary of the union for the latter—coarse models are specified in section 3.1.

Interestingly, maximizing the boundary surface of the selection is an indirect way to ascertain some diversity, since the overlap between conformers is minimized. Notice, though, that as opposed to the volume, the boundary surface area is not a monotonic function of the number of conformers. That is, for two selections S_1 and S_2 with $S_1 \subset S_2$, one has $volume(S_2) \geq volume(S_1)$, a property that may not hold for the boundary surface area.

2.4 The Greedy Strategy

2.4.1 The Strategy and its Guarantees

To solve our optimization problems, an obvious approach is the greedy strategy. The greedy strategy performs s steps, selecting at each iteration an element C_j of $\mathcal{C}$, that has not yet been selected, and that maximizes the sum of the weights of the cells being added. In other words, at each step, the algorithm selects a C_j that maximizes the weight of the union of the C_j.

Unfortunately, the selection obtained this way may not realize the optimum solution. As an example consider Fig. 5: For selecting two sets, the optimum choice has a weight of 14 whereas the greedy method yields a collection with a weight of 12. To quantify this performance, one resorts to the approximation ratio, that is the worst-case ratio between the solution returned and the optimal one. For max-k cover, this ratio is known to be of $1 - 1/e$, and is tight [CFN77, NWF78, FG89, Fei98].

Figure 5 Selecting two sets out of $C_1, \ldots, C_5$: the greedy strategy scores 12 (selecting C_3 and then C_2) while the optimum is 14 (selecting C_4 and C_5). The shaded cells have the weights as indicated and the unshaded cells have null weights.

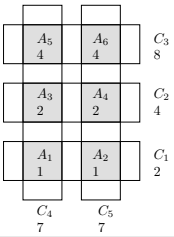
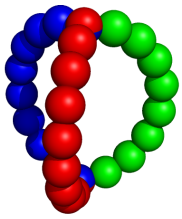


Figure 6 Three conformers of a fictitious protein loop: The surface of the union of the three conformers defines a double torus, i.e. a surface of genus two. Compare to Fig. 3(b).



2.4.2 Application to Conformer Selections

Volumetric decomposition, general weight w. Consider a volumetric decomposition as specified in section 2.1. The weighting scheme is called non-negative provided all weights are ≥ 0. The approximation ratio of the greedy strategy and its optimality are usually proved in the uniform weight case [CFN77, NWF78, FG89, Fei98]. The following theorems, proved in Appendix 7, generalize to non-negative weights:

Theorem 2.1. *Consider a volumetric decomposition with non-negative weights. For Problem 1, the greedy approach has an approximation ratio of* $1 - (1 - 1/s)^s > 1 - 1/e$.

Theorem 2.2. *The greedy approach cannot perform better than* $1 - (1 - 1/s)^s$.

Surface decomposition, boundary surface weight w_S. For volumetric decompositions, the previous bound indicates that one is always above 63% $(1 - 1/e)$ from the optimum. Unfortunately, as shown in Appendix 8.1, such a result does not hold for problem 2:

Observation 1. *Consider a surface decomposition. For Problem 2, the greedy approach may have a worst-case approximation ratio as bad as* $1/s^2$.

Practical considerations. In the following, we shall focus on the problem of optimizing the surface area rather then the volume of a selection for two reasons. First, we are not aware of any robust implementation to report the volume of a union of balls, whereas, robust and optimized algorithms exist to handle surface arrangements [CL07, CL, CCCLT].

3 Material and Methods

In this section, we introduce the models, concepts and algorithms used to make a geometric and topological assessment (section 4), and a docking assessment (section 5) of the selections.

3.1 Data Sets and Conformer Generation Methods

Protein models and their comparison. We use two protein models: the atomic model and the coarse model. Following [Zac03], given an atomic model, a coarse-residue-based model is obtained by replacing each side-chain by one or two pseudo-atoms, depending on the amino-acid type—the location of the C_α carbon does not change. To distinguish two models of a complex, say 1BTH, we shall use the notation 1BTH-atomic and 1BTH-coarse for the all atom and coarse model respectively.

A classical statistic used for comparing two conformations of the same protein is the C_α-rmsd, that is the standard deviation of the distance between the atomic positions of the C_α carbons of the two proteins. (Below, we shall use the C_α-rmsd to specify algorithm `HClust`, and to compare the selection of `Greedy` and `HClust`.)

While the C_α-rmsd is a good measure to compare two conformations of the same (portions of a) protein, a finer statistic is called for to evaluate the interface of a putative complex proposed by a rigid docking algorithm. To do so, we shall use the interface RMSD, denoted I-rmsd. To define it, call the two partners of the complex the ligand and the receptor, and assume that the receptor of the co-crystallized complex has been aligned with

that of the putative receptor. The I-rmsd is the C_α-rmsd restricted to selected atoms of the ligand: those identified in the native complex within a distance threshold of 7Å from the receptor [LMW07, BPZ06].

Protein loops: casting. We study four flexible protein loops belonging to the protein-protein interface of four complexes, 1OAZ, 1CGI, 1BTH and 3HHR. For each complex, both the unbound and the bound i.e. co-crystallized structures of the partners are known, and the conformation of the studied loops differs between these two forms. Three of the complexes (1CGI, 1BTH and 3HHR) come from the non-redundant protein-protein docking benchmark [CMJW03].

The four flexible loops differ by size and degree of variation between the bound and unbound forms, as characterized by the C_α-rmsd between the bound and unbound forms. In complex 1BTH, the 10 amino acid (aa) loop of the thrombin mutant bound to the pancreatic trypsin inhibitor undergoes a 5.7 Å deviation; in complex 1CGI, the structure of the 11 aa loop of α-Chymo-trypsinogen bound to pancreatic secretory trypsin inhibitor has not been resolved in the unbound form, showing a high degree of flexibility; in 1OAZ, the 12 aa loop of the Ige Fv Spe7 protein complexed with a recombinant thioredoxin only undergoes a 2.1 Å deviation, while in 3HHR, the 26 aa loop of the human growth hormone bound to the extracellular domain of its receptor presents a deviations of 5.5 Å.

Protein loops: docking difficulties. In fact, these test-cases are representative of different difficulty levels for docking systems containing flexible loops, found in the non-redundant protein-protein docking benchmark [21] and among CAPRI targets.

The loops can undergo limited amplitude, quasi-harmonic movements like 1OAZ. In that case, their deformation may be addressed by methods specific to low amplitude deformation, for example by exploring low amplitude normal modes of deformation during docking. See [Zac05] and [**?**] for the case of Capri target T18.

Large amplitude loop deformations as found in complexes 1BTH or 1CGI generally involve backbone refolding. Therefore, they cannot easily be searched during the docking process. These deformations count for the most difficult docking cases as also illustrated by CAPRI target T20 that features large loop deformation [LMW07]. Finally, long flexible loops, more than 15 amino acid long, such as 3HHR, present the additional difficulty to span a highly extended conformational space that cannot be exhaustively searched. Information on the volume compatible with partner association, obtained by a multi-copy docking process with highly diverse confomers, can help restricting the search of internal conformations and turn it tractable.

Conformer generations methods. A number of methods exist to generate atomic loop geometries [XSH02, NOS05, DYM$^+$07, SBL07, FDŠ00]. We selected `Direx` [SBL07] and `Loopy` [XSH02], which respectively generate dense and sparse (exploring more space) ensembles of conformers. (For completeness, an overview of these two methods is presented in Appendix 9.1.)

To quantify the diversity of the loop ensembles generated, we computed the MSA of the union of a collection of $n = 500$ conformers for the four models. (The residues involved in the MSA calculation are those from the loops together with the two residues bounding the loop, which are shared by all conformers.) To see that the `Loopy` data set is less redundant and explores more space, observe that the ratio $MSA(\texttt{Loopy})/MSA(\texttt{Direx})$ spans the range $[1.79, 4.16]$ and $[1.86, 4.60]$ respectively. See Table 2 in Appendix for a full report.

Geometry versus energy. Conformer generation methods, when applied to flexible loops, disregard the geometry of the scaffold accommodating the loop. To avoid steric clashes within the loop and between the loop and its scaffold, we computed the potential energy of the system *loop+scaffold* by 100-steps of energy minimization with `GROMACS` [LHvdS01]. We then discard two types of conformers: those featuring a *large* short range Lennard-Jones term, which witnesses steric clashes between the loop and the scaffold, and those with a *large* bonded energy, which witnesses clashes within the loop. More precisely, in each case, we discarded loops whose energy was larger than $Q_3 + 1.5(Q_3 - Q_1)$, with Q_i is the ith quartile. (A standard outlier filtering strategy [SB85] using box-and-whisker plot [CCKT83].)

3.2 Greedy Selection: Implementation

The naive and priority-based versions. Denote I_i the selection of i conformers after i steps of the greedy strategy, and let R_i stand for the remaining candidates. Following Eq. (2), the naive way of computing I_i consists of incrementally linearly scanning all possible solutions, that is

$$I_i = \arg \max_{C_j \in R_{i-1}} w(I_{i-1} \cup \{C_j\}). \tag{3}$$

As proved in Appendix 8, the following complexity is worst-case optimal:

Theorem 3.1. *Consider n conformers of a molecule made up of b balls. To select s conformers, the naive version of algorithm* `Greedy` *has complexity $O(b^2 n s^3)$.*

A more elaborate strategy than the naive (brute force) method maintains the increments associated to all candidates, so as to select the best one from a priority queue. To do so, one needs in particular the surface arrangements on all spheres, together with the inclusion information of spherical patches into the other conformers. To account for this information, which encodes the complexity of the surface arrangement, define

$$\tau = \sum_{C_i \in \mathcal{C}} \sum_{S_j \in C_i} \sum_{P_k} \mathbf{1}_{S_j \text{ covers patch } P_k \text{ or } P_k \text{ lies on } S_j}, \tag{4}$$

where $\mathbf{1}_X$ is the characteristic function of the Boolean variable X, $\mathcal{C}$ is the set of all conformers, S_j a sphere of a conformer C_j and P_k a patch on a sphere of a conformer in $\mathcal{C}$.

This *priority-based* version, presented in Appendix 8, satisfies:

Theorem 3.2. *The priority-based version of Algorithm* `Greedy` *has amortized complexity* $O(\tau + s \log n)$.

Implementations. The naive version is implemented using the `Delaunay_3` and `Alpha_shape_3` packages of the Computational Geometry Algorithms Library [CGA]. The priority-based version is implemented using the surface arrangements package described in [CL07, CL, CCCLT], which is the only one, to the best of our knowledge, able to compute effectively the exact arrangement of circles on a sphere. In both cases, robustness issues are critical due to the density of conformers manipulated.

3.3 Conformer Selection Methods

We compare algorithm `Greedy` against one contender, Algorithm `HClust`, which is a hierarchical agglomerative clustering [Gor99] method based on the *average linkage*, used for protein-protein docking [BPZ06]. (We also tested the single linkage and complete linkage strategies, which performed equally w.r.t. the MSA—data not shown.) Given a dissimilarity measure between two clusters (i.e. groups of conformers), the algorithm generates a binary tree encoding a sequence of nested partitions of the n conformers. Notice that the coarser partition features one cluster containing the n conformers, while the finer partition features n clusters of a single conformer. As dissimilarity measure, we use the C_α-rmsd between pairs of conformers. Cutting this binary tree at an appropriate level provides the number of desired conformers, since we select one representative within each cluster. The representative selection was carried out through a two-stage process, namely (i) a fictitious *average* loop is computed: for k conformers each consisting of b balls centered at $c_{i,j}$, with $i = 1, \ldots, k$ and $j = 1, \ldots, b$, the fictitious loop consists of b balls centered at $\overline{c_j} = (\sum_{i=1,\ldots,k} c_{i,j})/k$; (ii) the representative is taken as the conformer from the cluster having the least C_α-rmsd with this fictitious loop.

Note that a sequence of selections of increasing size provided by `HClust` are not nested. One indeed gains one conformer by splitting one cluster K (corresponding to a node n_K in the binary tree) into two clusters K_1 and K_2 (the sons of node n_K in the binary tree). But the representative conformer C_i of cluster K may not be that of the cluster (K_1 or K_2) the conformer C_i belongs to.

4 Diverse Ensembles: Geometric and Topological Assessment

In this section, we discuss geometric and topological quantities to characterize the diversity of an ensemble, and compare those produced by the `Greedy` and `HClust` algorithms on four protein models.

4.1 Statistics of Interest: Geometry vs. Topology

Comparing MSA. We first use the MSA of the union of conformers selected by either `Greedy` or `HClust` as a diversity criterion of the selection. To see how, for a given selection method M (G: greedy; H: hierarchical), let $\mathcal{N}_M = \{I_1, \ldots, I_n\}$ be a *collection of selections* of increasing size, i.e. selection I_i contains i conformers. Note that `Greedy` provides a *nested* collection of selections, since the selection I_{i+1} of size $i+1$ is the selection I_i of size i to which an additional conformer has been prepended, while `HClust` does not.

To compare two collections of selections, both for the atomic and the coarse models, we report two sets of values. Let R_M be the maximum MSA obtained over all selections in $\mathcal{N}_M$, that is $R_M = \max_{I_i \in \mathcal{N}_M} MSA(I_i)$. First, we focus on the maxima of MSA reached, and report R_G/R_H. Second, denote n_{H_x} the smallest number of conformers required by `HClust` to get a MSA (say A) equal to $x\%$ of its maximum. Then, denote n_G the least number of conformers required by `Greedy` to get a MSA greater or equal to A. We report n_{H_x}/n_G for $x = 100\%$ and $x = 95\%$.

Comparing the topology. Apart from the MSA, an interesting information about the selection is the topology of the union of the balls of the conformers selected. The boundary of the union of conformers defines a compact orientable surface, possibly non-connected—as the union of conformers may isolate one or several hole(s). By the theorem of classification of connected compact orientable surfaces [Hen94], each such connected component is a sphere with a number $g \geq 0$ of handles attached: for example, the sphere, one-torus, two-torus respectively correspond to $g = 0, g = 1, g = 2$. To characterize these situations, one resorts to Betti numbers, which are respectively $\beta_0 = 1, \beta_1 = 2g, \beta_2 = 1$. Alternatively, one can compute the Euler characteristic of the surface, that is $\chi = \beta_0 - \beta_1 + \beta_2 = 2 - 2g$, with g the genus of the surface. Fig. 6 presents and example selection of $g+1$ conformers anchored at the loops extremities, and defining a genus g surface ($g = 2$ here). We shall compare the variation of β_1 for $n_{G_{100\%}}$ conformers selected by algorithm `Greedy` and `HClust`.

Comparing the C_α-rmsd. The measures just described are somewhat tailored to our selection algorithm, since `Greedy` aims at maximizing the MSA. To provide a fair comparison, we thus also report on a measure based upon the C_α-rmsd used by `HClust`. More precisely, to make an assessment on the diversity of a given selection, we investigate the range spanned by the C_α-rmsd of loops from this selection with respect to the native co-crystallized loop. Notice that since the C_α carbons are common to an atomic model and its coarse representation (see beginning of section 3.1), algorithm `HClust` reports the same selection for the atomic and coarse models, while algorithm `Greedy` reports two different such selections.

4.2 Results

Comparing MSA. In the following, we refer to Tables 3 and 4 in Appendix 9.2. Speaking of the max values R_G and R_H, one observes that `Greedy` yields an increase in the range 9-13% for (`Direx`, atomic), 11-15% for (`Direx`, coarse), 14-54% for (`Loopy`, atomic) and 25-56% for (`Loopy`, coarse). As seen from the ration $n_{H_{100\%}}/n_G$, the number of conformers required by algorithm `Greedy` to match the maximum of algorithm `HClust` incurs a dramatic k-fold reduction, where k spans the following ranges (decimals omitted): 9-154 (`Direx`, atomic), 1-160 (`Direx`, coarse), 4-79 (`Loopy`, atomic), 10-79 (`Loopy`, coarse). On the other hand, as can be seen from the plot Fig. 7 (a typical one), the asymptote is reached rather fast for both algorithms. Focusing on 95% of the max MSA obtained, the ratios $n_{H_{95\%}}/n_G$ now span the following ranges: 1-6 (`Direx`, atomic), 1-3 (`Direx`, coarse), 3-28 (`Loopy`, atomic), 3-11 (`Loopy`, coarse). These values call for two conclusions.

First, consider the variation of the ratio $(n_{H_{100\%}}/n_G)/(n_{H_{95\%}}/n_G)$ for the `Direx` and `Loopy` data sets. This ratio is clearly much higher for `Direx` than `Loopy`, which has the following explanation: for a dense data set such as `Direx`, algorithm `HClust` selects pretty fast good representatives accounting for most of the MSA (95% here); but further selections fail at significantly increasing the MSA, as seen from much higher ratios $n_{H_{100\%}}/n_G$. On the other hand, algorithm `Greedy` consistently selects the conformers optimizing the increase of MSA. Second, focus on the statistic $n_{H_{95\%}}/n_G$ for the `Direx` and `Loopy` data sets. This ratio is much higher for the latter data set, which shows that algorithm `Greedy` is also better at selecting large increments of MSA within data sets of conformers exploring more space.

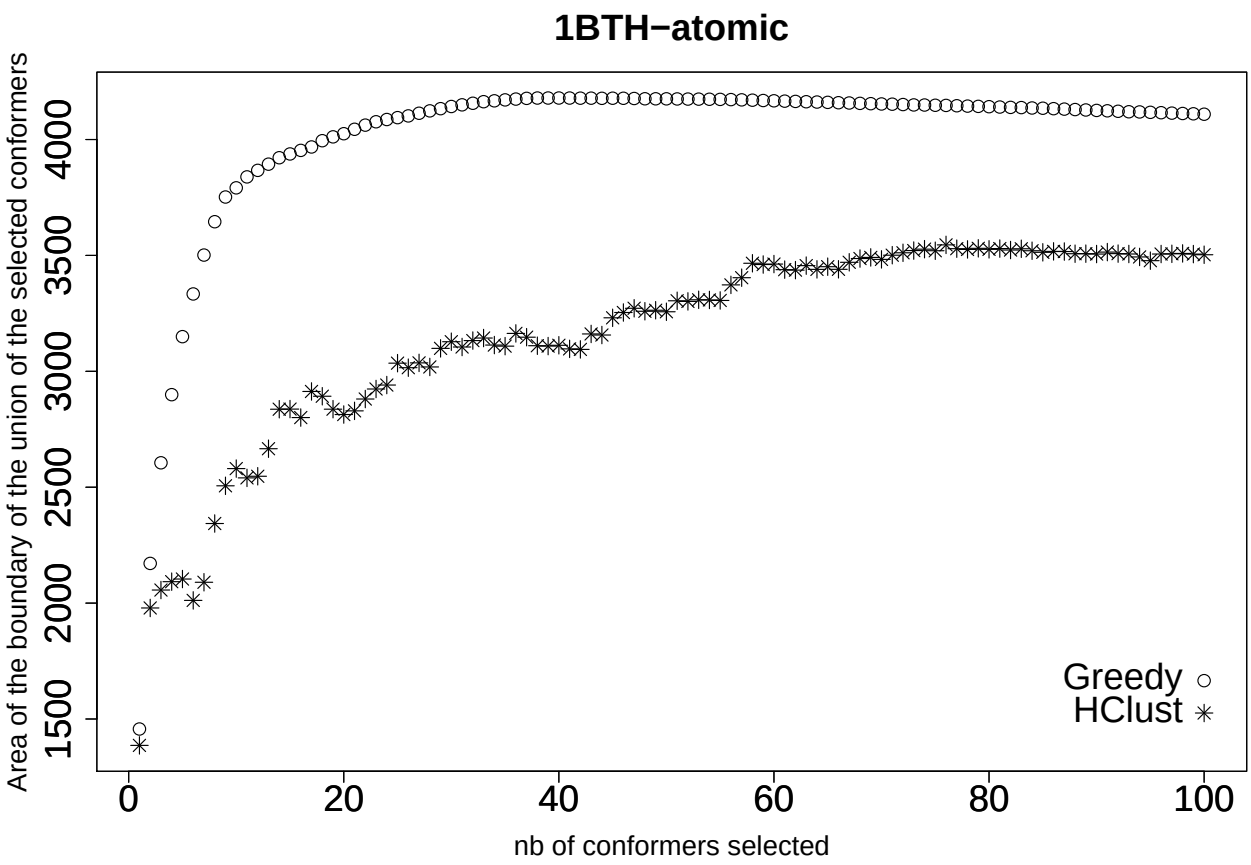
Variations of Betti numbers. For a qualitative explanation of these facts [1], consider the variation of the first Betti number β_1 for the two algorithms. As seen from Tables 5 and 6 in Appendix 9.2, the selection obtained with algorithm `Greedy`, when compared to that obtained with `HClust`, typically features an average value of β_1 which is about 12 times higher for `Direx` and 5 times higher for `Loopy`.

The variation of β_1 is illustrated on Fig. 8, which is also a typical plot. Indeed, all such curves feature a sharp peak, followed by a plateau, and algorithm `Greedy` outperforms its contenders in both regimes. The sharp rise at the beginning of the selection process corresponds to the choice of *independent* conformers i.e. conformers that do not overlap excepted at their extremities. Such conformers minimize the overlap between balls—in agreement with the criterion targeted by algorithm `Greedy`. Once the maximum has been reached, the conformers selected bridge gaps, whence a decrease in β_1. The sharp decrease stops as soon as the union of the selection is *essentially* a topological ball. The union still features small handles. Such handles get created and destroyed upon addition of new conformers, whence the minute fluctuations about the horizontal asymptote of the graphs displaying the variation of β_1.

The variation of β_1 (see plots in Appendix 9.3) also sheds an interesting light on the relative flexibility of the four loops. As for the MSA variation, the curve of 3HHR clearly shows that the $n = 500$ conformers are

[1] The analysis is qualitative for the following reason: the contribution of a handle to β_1 is the same whatever its size. That is a large handle coming from a whole loop (as on Fig. 6) has the same weight as a small one coming from the creation of a local cycle between atoms of say a side-chain and the backbone. Computing the Betti numbers is by now standard—we use the α-shapes-based algorithm of [DE95]. But the calculation of a geometrically pleasant basis of the homology groups is still an active area of research [CF07].

Figure 7 `Loopy` data set. Variation of MSA with the selection size: conformer selections of `Greedy` always expose more surface than those of `HClust`, a diversity criterion.



not enough in the `Loopy` data set. We also speculate that a comparison between the maximum value of β_1 obtained and the ensuing plateau encode interesting features on hinges found in the structure. To confirm these statements, though, one would need to geometrically qualify the geometry of the handles defining a basis of the homology groups.

Comparing the C_α-rmsd. Let δ_G and δ_H be the range of C_α-rmsd with respect to the native bound loop spanned by the conformers from the selection achieved by `Greedy` and `HClust` respectively. To compare the merits of the two contenders, we plot the variation of $(\delta_G - \delta_H)/\delta_H$ as a function of the selection size. See Fig. 9 and tables 7 and 8 for the values. (The selection size on this figure stops at 30 for two reasons: (i) for larger selections, the discrepancy $\delta_G - \delta_H$ rapidly goes to zero, and (ii) typical selection sizes for docking are below 30 anyway.)

For models 1BTH, 1CGI and 1OAZ, apart from exceptions, `Greedy` outperforms `HClust` regardless of the data set and of the representation level, even though the C_α-rmsd is not the criterion targeted. For 3HHR, while the two algorithms perform almost equally for the `Direx` data set, `Greedy` is clearly outperformed for the `Loopy` one. This owes to the length of the loop and its flexibility. In particular, the fact that the same extreme conformers (w.r.t. the native loop) get selected whatever the selection size, whence a constant ratio $(\delta_G - \delta_H)/\delta_H$, points at an insufficient pool of conformers.

Figure 8 `Loopy` data set. Variation of the first Betti number β_1 with the selection size: more handles are observed for conformer selections of `Greedy` than for those of `HClust`.

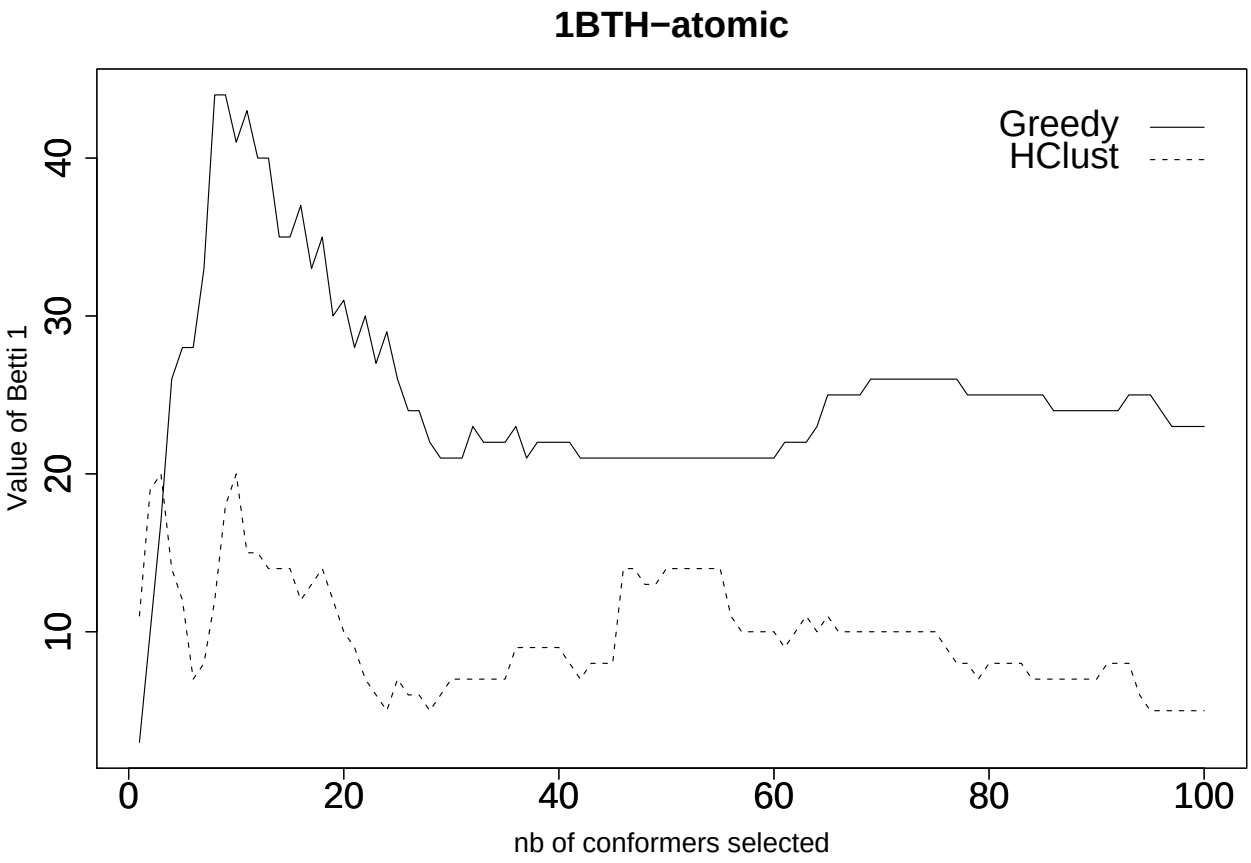
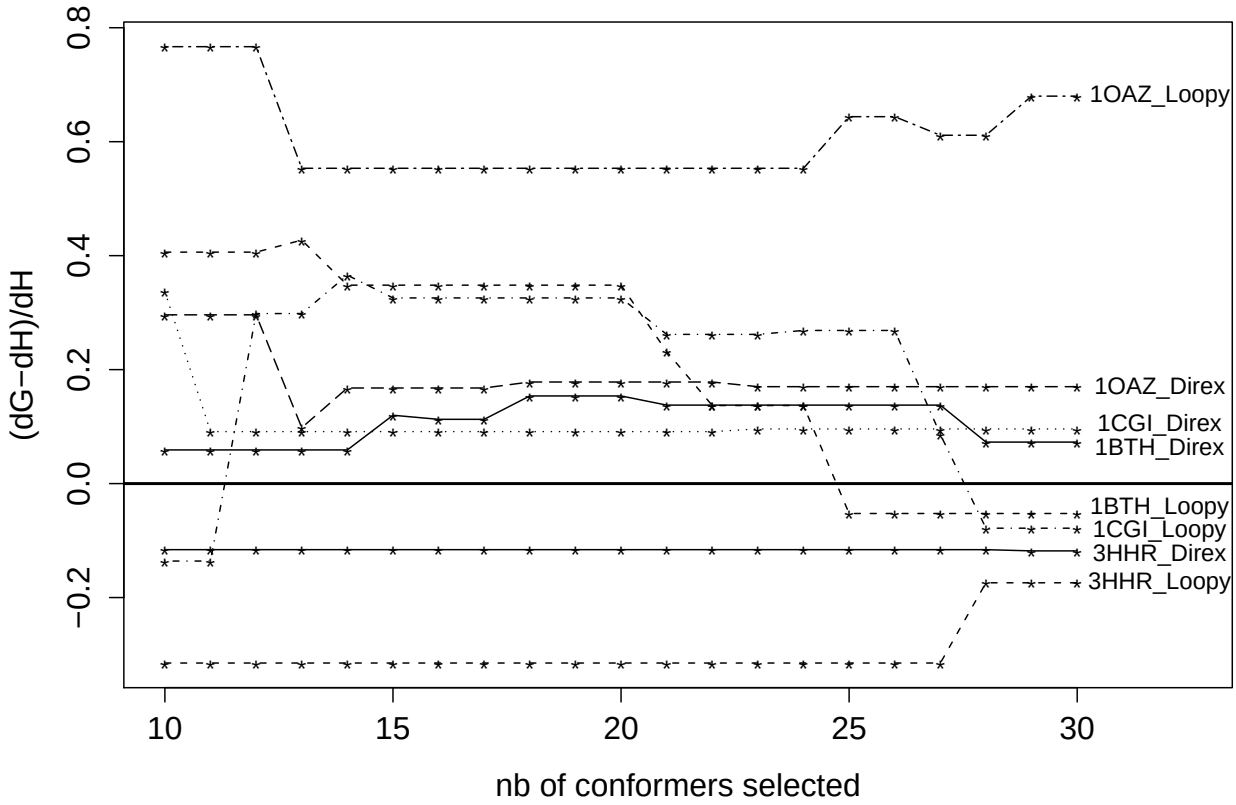


Figure 9 Comparing ranges of C_α-rmsd w.r.t. the native loop for atomic models: plotting $(\delta_G - \delta_H)/\delta_H$. See text for details.



5 Diverse Ensembles: Docking Assessment

In this section, we report docking results for `1BTH`,`1CGI` and `1OAZ`, based on selections of coarse conformers provided by algorithms `HClust` or `Greedy`. Following the discussion in section 4, we focus on conformer pools generated by `Loopy`, which are more diverse, and we omit complex 3HHR, since generating a representative pool of conformers for its long flexible loops of 26 amino is a problem in itself.

5.1 Docking as a Multi-scale Process

Docking is a complex process requiring in general a multi-scale approach, and our conformer selection strategy is meant to occur at early stages. More precisely, we place ourselves at the starting point of a docking simulation, where putative docking geometries are detected for further refinement. Our point is to know whether a given interface presenting a flexible loop can be a possible binding interface, in which case it is worth spending time on higher resolution exploration, for example using MC or MD simulations. To this aim, we work with a limited number of conformers to represent the flexible fragment. As a matter of fact, all unsuccessful or poor quality predictions of the last CAPRI rounds concern targets that present from limited to high amplitude remodeling of flexible interface fragments []. These fragments consist of one or several loops (targets 18, 20, 26), terminal helices (targets 1, 24), or domains (target 28). For all these cases, the correct geometry of association is either not generated or not retained after initial 3D-space exploration.

5.2 Docking Protocols

Specifying the ligand and the receptor. We ran docking simulations on three complexes to validate the conformer selection strategy based upon MSA maximization. Each complex was decomposed into one rigid

Table 1 Percentage of the maximum MSA achieved by `Greedy` realized by the selection of only ten loops. For all models but 3HHR, a small number of conformers yields (almost) the maximum of the MSA.

`Loopy`	%	`Direx`	%
1BTH-atomic	90.73	1BTH-atomic	99.33
1CGI-atomic	95.73	1CGI-atomic	100
1OAZ-atomic	82.85	1OAZ-atomic	99.77
3HHR-atomic	66.46	3HHR-atomic	99.74
1BTH-coarse	87.72	1BTH-coarse	98.88
1CGI-coarse	91.23	1CGI-coarse	99.7
1OAZ-coarse	83.12	1OAZ-coarse	99.44
3HHR-coarse	55.03	3HHR-coarse	99.99

protein called the ligand (L), and one flexible called the receptor. The receptor itself decomposes into a rigid template (T) and a flexible loop (F). While performing flexible protein docking with conformer ensembles, the strategy consists of using a conformer ensemble for the flexible loop F, this ensemble being selected from a larger pool. Thus, specifying a docking protocol requires specifying the triple T/L/F.

To see how, recall that a binary complex used for docking validation features two molecules which have been crystallized under two forms: on their own, i.e. the unbound forms, and in complex i.e. the bound forms. Thus, to specify the rigid parts (T and L), we provide a tag indicating the origin of the partner, namely U for Unbound and B for Bound. To specify the ensemble associated to F, we provide three pieces of informations: (i) the bound/unbound tag which indicates the loop geometry used to generate the pool of conformers (ii) the algorithm used to select the conformers from this pool (`HClust` or `Greedy` here), and (iii) the selection size. For example, F=B-`Greedy`-10 refers to 10 conformers selected by algorithm `Greedy`, out of a pool of conformers generated from the Bound structure of the receptor. As a second example, F=B-1 means that a single loop has been used, the Bound one.
To summarize, we report on the following six docking protocols: three using the Bound form of the receptor, namely B/B/B-1, B/B/B-`HClust`-10, B/B/B-`Greedy`-10; and three using the Unbound form of the receptor, namely U/B/B-1, U/B/B-`HClust`-10, U/B/B-`Greedy`-10.

Two comments on these protocols are in order. First, notice that the incentive for using the Bound conformation of the flexible region to generate the conformers is the following: for very flexible systems, such as 1CGI mentioned in section 3.1, the reconstruction of the unbound conformation of the flexible loop from the crystallographic data is not possible. (If the conformation of the loop changes across the crystallographic units, the signal is not strong enough for the reconstruction to be carried out.) Second, the particular protocols B/B/B-1 and U/B/B-1 can be seen as sanity checks, since in using only the native loop conformer, one expects the docking process to yield satisfactorily complexes.
About the pool size and the number of conformers. For each flexible loop, a pool of $n = 500$ conformers was generated using `Loopy` [XSH02], from which $s = 10$ were selected using the `Greedy` and `HClust` algorithms. Following [BPZ06], the choice of $s = 10$ comes from a trade-off between the requirement to have a representative selection, and the computational resources. As seen from Table 1, we observe that for all systems but `3HHR`, the MSA of the union of the first 10 conformers selected by `Greedy` realizes more than 80% of the maximum MSA observed along the iterative greedy selection up to n conformers. The same table shows that a mere 10 conformers is not enough to represent the flexible loop of `3HHR`.

5.3 Initial Conditions for a Protocol

For a given protocol, we ran N_t docking tests using algorithm `ATTRACT` [Zac03], which is based on the coarse protein representation recalled in section 3.1. This algorithm has been adapted to handle multiple copies of a flexible loop in [BPZ06]. In this scheme, using Boltzmann's principle, each copy is assigned a fitness score (between 0 and 1) based upon its interaction energy with the receptor. Each docking test corresponds to a specific position and orientation of the ligand with respect to the receptor. Given these initial conditions, ATTRACT performs a sequence of minimizations so as to explore the six degrees of freedom of the ligand. At each stage, the energy of each conformation of the complex is computed. Upon termination, the loop selected is that having the highest fitness score. An assessment of the quality of the proposed complex is then based upon two figures: (i) the interaction potential energy E of the complex (ii) the I-rmsd of the atoms of the ligand.

For the N_t tests associated to a given protocol, the plot of the pairs (E,I-rmsd) defines the energy landscape of the docking experiment. Thus, a conformer ensemble is satisfactory if the landscape features at least one conformer yielding a large number of points (E,I-rmsd) next to the bottom left corner of the energy landscape. Practically, we represent an energy landscape using buckets. For a given bucket B_i and conformer C_j, let $s_{i,j}$ be the number of times conformer C_j yields a complex whose energy and I-rmsd fall in bucket B_i. (Notice that $\sum_{i,j} s_{i,j} = N_t$.) Finally, for a given bucket B_i, denote l_i the index of the conformer that yields the largest value of $s_{i,j}$, and let $r_i = \sum_{j=1,\ldots,n;j\neq l_i} s_{i,j}$. In bucket B_i we display the score s_{i,l_i}, together with r_i when $r_i \neq 0$. The color used is that associated to conformer l_i, with one color per conformer.

5.4 Results

For each selection method, a total of $N_t \sim 35,000$ docking tests were run using the same $s = 10$ selected conformers. To analyse the results, we plot the portion of the energy landscape corresponding to a I-rmsd ≤ 15Å with a negative energy. An example of such a plot is presented on Fig. 10, and we refer the reader to Appendix 9.4 for the remaining plots. In analyzing landscape, since the docking process is coarse, we just aim to identify conformers with good potential for atomic docking process. We thus skip a detailed atomic discussion of the results, a notoriously difficult task [LSF08]. For six docking protocols examined (three systems, Bound and Unbound receptors for each), we argue that the results decompose as follows: three favorable to `Greedy`, two ties, one favorable to `HClust`.

Docking improved using `Greedy`. For complex `1BTH`, the docking protocol B/B/B-`Greedy`-10 leads to 161 predictions with I-rmsd $\in (1;2]$ and energy below -21 units. Only 3 such predictions are found using docking protocol B/B/B-`HClust`-10. See Fig. 10. The same kind of result can be observed when using the unbound form of the receptor of `1CGI`. Indeed, U/B/B-`Greedy`-10 leads to 160 predictions with I-rmsd $\in (3;4]$ and energy below -15 units, while U/B/B-`HClust`-10 yields 18 such predictions. For complex `1OAZ`, neither U/B/B-`Greedy`-10 nor U/B/B-`HClust`-10 leads to a high number of predictions below 5Å I-rmsd and below -15 energy units: 5 with one loop, and 13 with 5 different loops respectively. But considering predictions with I-rmsd in interval $(5,7]$ and energy below -15 units, U/B/B-`Greedy`-10 leads to 195 predictions with the same loop while U/B/B-`HClust`-10 leads to one such prediction.

Tie between `Greedy` and `HClust`. The results of the docking involving the unbound form of the receptor of the complex `1BTH` are more ambiguous. U/B/B-`HClust`-10 leads to two predictions with the same loop with I-rmsd below 5 Å and below -15 energy units, while U/B/B-`Greedy`-10 leads to no such prediction. When considering predictions with I-rmsd in interval $(5,7]$ and energy below -15 energy units, both U/B/B-`Greedy`-10 and U/B/B-`HClust`-10 yield about 150 predictions. The results of the docking involving the bound form of the receptor of the complex `1OAZ` needs further scrutiny to detect whether some improvement is achieved by B/B/B-`Greedy`-10 compared to U/B/B-`HClust`-10. Indeed, no highly populated region with low energy and low I-rmsd clearly emerges.

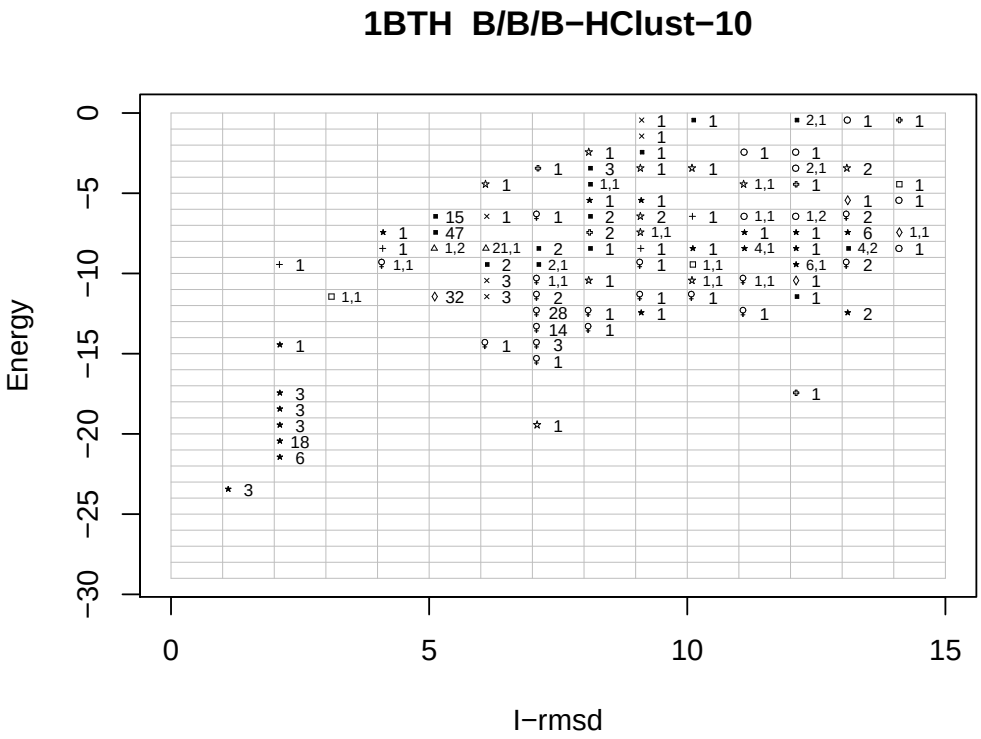
No improvement while using `Greedy`. The results of the docking involving the bound form of the receptor of the complex `1CGI` are more favorable to the docking protocol B/B/B-`HClust`-10. Nevertheless, it must be noticed that even if B/B/B-`HClust`-10 leads to a larger number of good predictions, the energy of these predictions is much higher than those obtained with the native loop in the protocol B/B/B-1.

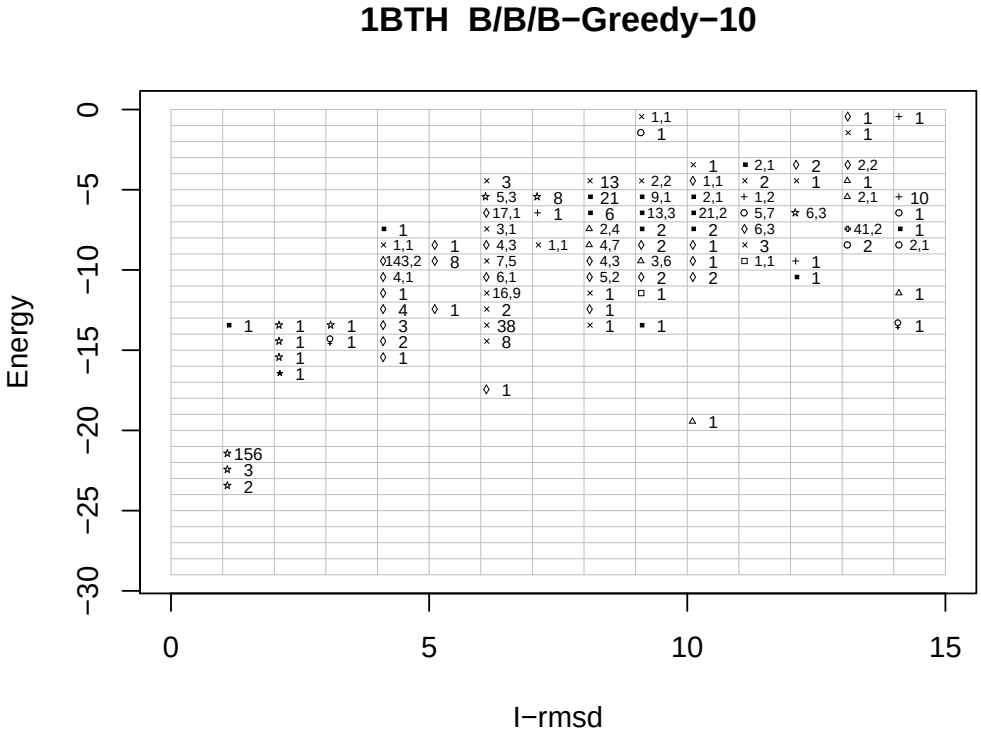
5.5 Running Times

Some comments are in order regarding the computational cost of the docking and selection algorithms. The naive and priority-based selection algorithms were run on a PC computer equipped with a Xeon processor (quadcore) at 2.33GHz, and 16GB of RAM. Surprisingly, we observed a factor of one order of magnitude in favor of the naive implementation, and in particularly when s is much smaller than n. Although asymptotically optimal, the problem of the priority-based algorithm lies in the computation of the arrangement: for n conformers, the size of the arrangement on a given ball may be (and actually is on some examples) as high as n^2. Using the naive implementation, the selection of ten loops requires about half an hour using the all atoms representation, and five minutes using the coarse grain representation.

As reported in [BPZ06], the docking algorithm using ten conformers requires 31 hours on a 2.2 GHz Athlon PC.

Figure 10 `1BTH`: binning the docking tests using the Bound form of the receptor. The color associated to a bucket corresponds to the loop yielding the highest score in this bucket. A highly populated region next to the bottom left corner of the plot indicates that a satisfactory conformer was present. See text for details.





6 Conclusion

Summary of results. For systems whose flexibility cannot be explored resorting to molecular dynamics simulations, the manipulation of discrete ensemble of pre-generated conformers is the route of choice. This strategy is valid for fragments of any size, namely for side chains, protein loops or domains. Because the generation of such ensembles does not take into account the whole environment of the fragment (in the whole protein or complex), the energetic functionals used to compute the energy of a conformer cannot, in general, be directly related to the thermodynamic equilibrium between the conformations. This observation calls for the development of methods providing a rather uniform sampling of the conformational space of the fragment considered, so as to retain conformers avoiding obvious steric clashes. But such algorithms face one central difficulty: that of characterizing the conformational space coverage, so as to maximize the diversity of the conformers. In this context, we make three contributions.

First, we present geometric optimization methods geared towards the characterization and the selection of conformational diversity. Given a collection of conformers, the methods aim at returning a selection maximizing a functional of the volume occupied by the conformers, or of the molecular surface exposed by the conformers. Greedy strategies are used to solve these problems, and theoretical bounds are proved.

Second, for the particular problem of the optimization of the MSA, we make a geometric assessment of the conformational diversity of the conformers selected, based upon experiments carried out on four flexible protein loops. We show that our greedy strategy matches the MSA of standard selection methods, using, depending on the particular system and the model (atomic or coarse), a number of conformers between *one and two orders of magnitude* smaller. Moreover, tracking the variation of the MSA together with topological informations of the selection (the Betti numbers) yields insights on the quality of the coverage of the conformational space associated to a collection of conformers.

Third, using coarse representations of three of these protein models, we compare the results of a multi-copy docking algorithm, for two sets of copies: one selected by our greedy strategy—`Greedy`, and one generated by a standard hierarchical clustering algorithm—`HClust`. For six docking protocols (three systems, Bound and Unbound receptors for each), the results decompose as follows: three favorable to `Greedy`, two ties, one favorable to `HClust`.

Applications and outlook. Our developments have a number of direct applications. First, our characterization of the conformational diversity based upon geometric and topological measures, together with the greedy strategy, should prove useful to improve the conformational space coverage of conformer generation methods. For example, algorithms `Loopy` and `Direx` could bootstrap on our selections so as to improve their conformational diversity. Second, the positive results obtained for coarse docking call for further developments. In particular, bootstrapping on the selections of coarse conformers generated by `Greedy` so as to generate high-quality atomic models should improve the predictions for challenging flexible protein-protein complexes.

Interestingly, our work also raises a number of open theoretical questions. First, for a particular problem (conformer generation, docking), the question of the particular functional to be optimized (volume based, surface based) needs to be addressed. Volume based and surface based are obvious candidates, especially since the surface exposed by a collection of balls is the *geometric locus* where interaction occurs. But these might be seen as a *first approximations* to qualify the conformational diversity. That is, because covering a $3D$ volume with a collection of conformers does not admit a unique solution, it might actually be necessary to incorporate into the functional some measure of the multiplicity of the cells of the volume or surface arrangements, so as to guarantee that each portion of space is covered the same number of times. Second and from a more algorithmic perspective, while our current running times are comparable to those required by the algorithms exploiting the conformer selections, provably good output sensitive algorithms deserve further investigation.

Acknowledgments

The authors wish to thank J. Bernauer, G.F. Schröder, P. Yao on the one hand, and F. Nielsen on the other, for insightful discussions about conformer generation methods and optimization, respectively. The molecular illustrations were produced using the software PyMOL [W.L].

References

[AE96] N. Akkiraju and H. Edelsbrunner. Triangulating the surface of a molecule. *Discrete Applied Mathematics*, 71(1-3):5–22, 1996.

[BPZ06] K. Bastard, C. Prévost, and M. Zacharias. Accounting for loop flexibility during protein-protein docking. *Proteins*, 62(4):956–969, 2006.

[CCCLT] Pedro M. M. de Castro, Frédéric Cazals, Sébastien Loriot, and Monique Teillaud. Design of the CGAL spherical kernel and application to arrangements of circles on a sphere. *Computational Geometry : Theory and Applications*. Under revision.

[CCKT83] J.M. Chambers, WS Cleveland, B. Kleiner, and PA Tukey. *Graphical Methods for Data Analysis*. Belmont, 1983.

[CF07] C. Chen and D. Freedman. Quantifying homology classes ii: Localization and stability. *Preprint*, 2007. arXiv:0709.2512v2.

[CFN77] G. Cornuejols, M.L. Fisher, and G. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.

[CGA] CGAL, Computational Geometry Algorithms Library. `http://www.cgal.org`.

[CL] F. Cazals and S. Loriot. Computing the arrangement of circles on a sphere, with applications in structural biology. *Computational Geometry : Theory and Applications*. Under revision.

[CL07] F. Cazals and S. Loriot. Computing the exact arrangement of circles on a sphere, with applications in structural biology. In *Proc. 23th Annual Symposium Computational Geometry—Video/Multimedia track*, 2007.

[CMJW03] R. Chen, J. Mintseris, J. Janin, and Z. Weng. A protein–protein docking benchmark. *Proteins*, 52(1):88–91, 2003.

[CSD03] A.A. Canutescu, A. A. Shelenkov, and R.L. Dunbrack. A graph theory algorithm for protein side-chain prediction. *Protein Science*, 12:2001–2014, 2003.

[CSRdA+05] J. Cortés, T. Siméon, V. Ruiz de Angulo, D. Guieysse, M. Remaud-Siméon, and V. Tran. A path planning approach for computing large-amplitude motions of flexible molecules. *Bioinformatics*, 21:116–125, 2005.

[DE95] C.J.A. Delfinado and H. Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 1995.

[dGvAS+97] BL de Groot, DMF van Aalten, RM Scheek, A. Amadei, G. Vriend, and HJC Berendsen. Prediction of protein conformational freedom from distance constraints. *Proteins Structure Function and Genetics*, 29(2):240–251, 1997.

[DYM+07] A. Dhanik, P. Yao, N. Marz, R. Propper, C. Kou, G. Liu, H. van den Bedem, and J.C. Latombe. Efficient algorithms to explore conformation spaces of flexible protein loops. In *7th Workshop on Algorithms in Bioinformatics (WABI), Philadelphia*, pages 265–276, 2007.

[FDŠ00] A. Fiser, R.K.G. Do, and A. Šali. Modeling of loops in protein structures. *Protein science*, 9(09):1753–1773, 2000.

[Fei98] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[FG89] PC Fishburn and WV Gehrlein. Pick-and choose heuristics for partial set covering. *Discrete Applied Mathematics*, 22(2):119–132, 1989.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.

[GLN04] R. Grunberg, J. Leckner, and M. Nilges. Complementarity of structure ensembles in protein-protein binding. *Structure*, 12:2125–2136, 2004.

[Gor99] AD Gordon. *Classification*. Chapman & Hall/CRC, 1999.

[Hen94] M. Henle. *A Combinatorial Introduction to Topology*. Dover, 1994.

[JW07] J. Janin and S. Wodak. The Third CAPRI Assessment Meeting Toronto, Canada, April 20–21, 2007. *Structure*, 15(7):755–759, 2007.

[LHvdS01] E. Lindahl, B. Hess, and D. van der Spoel. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *Journal of Molecular Modeling*, 7(8):306–317, 2001.

[LMW07] MF Lensink, R. Mendez, and SJ Wodak. Docking and scoring protein complexes: CAPRI 3rd Edition. *Proteins*, 69(4):704–18, 2007.

[LSF08] N. London and O. Schueler-Furman. Funnel Hunting in a Rough Terrain: Learning and Discriminating Native Energy Funnels. *Structure*, 16(2):269–279, 2008.

[MSLW08] A. Malevanets, F. Sirota-Leite, and S. Wodak. Mechanism and energy landscape of domain swapping in the B1 domain of protein G. *Journal of Molecular Biology*, To appear, 2008.

[MWC65] J. Monod, J. Wyman, and JP. Changeux. On the nature of allosteric transitions: a plausible model. *Journal Molecular Biology*, 12:88–118, 1965.

[NOS05] K. Noonan, D. O'Brien, and J. Snoeyink. Probik: Protein Backbone Motion by Inverse Kinematics. *The International Journal of Robotics Research*, 24(11):971, 2005.

[NWF78] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[SB85] G. Saporta and JM Bouroche. *L'Analyse des Données*. Presses Universitaires de France, Paris, 1985.

[SBL07] G.F. Schröder, A.T. Brunger, and M. Levitt. Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution. *Structure*, 15:1630–1641, 2007.

[W.L] DeLano W.L. The PyMOL molecular graphics system. `http://www.pymol.org`.

[XSH02] Z. Xiang, C.S. Soto, and B. Honig. Evaluating conformational free energies: The colony energy and its application to the problem of loop prediction. *Proceedings of the National Academy of Sciences*, 99(11):7432, 2002.

[Zac03] M. Zacharias. Protein-protein docking with a reduced protein model accounting for side-chain flexibility. *Protein Science*, 12(6):1271–1282, 2003.

[Zac05] M. Zacharias. Attract: Protein-protein docking in capri using a reduced protein model. *Proteins*, 60:252–256, 2005.

7 Appendix: Volumetric Decompositions

7.1 Greedy: Approximation Factor and Optimality

7.1.1 Approximation Factor

We shall use the following notation. The conformer selected at the kth step is denoted C_k, and the weight of the optimum set of conformers OPT. Also, let us denote by $w^*(C_k)$ as the sum of the weights of the new elements in C_k that have not been covered in $C_j, 1 \leq j < k$ (i.e. the weight increment at step k). We need the following lemma in order to prove theorem 2.1.

Lemma 7.1. *For $1 \leq k \leq s$, the following holds:*

$$w^*(C_k) + \frac{1}{s}\sum_{j=1}^{k-1} w^*(C_j) \geq \frac{OPT}{s}. \tag{5}$$

Proof. At the kth step, we select C_k that maximizes the weight of the new cells U_i being covered. The weight of the cells that are covered by the optimum solution but not yet covered by the $(k-1)$ is at least

$$OPT - \sum_{j=1}^{k-1} w^*(C_j) \tag{6}$$

Since w is non-negative, the union-bound property states that for any collection of conformers $C_1, \ldots, C_p$, one has $w(C_1 \cup \cdots \cup C_p) \leq \sum_{i=1,\ldots,p} w(C_i)$. Since all the cells involved in Eq. (6) are covered by the optimum set of conformers, by the union-bound property, there must exist one conformer,not yet selected, that covers these new cells with total weight at least

$$\frac{1}{s}\left(OPT - \sum_{j=1}^{k-1} w^*(C_j)\right). \tag{7}$$

Since C_k maximizes the weight of the new cells being covered, we must have

$$w^*(C_k) \geq \frac{1}{s}\left(OPT - \sum_{j=1}^{k-1} w^*(C_j)\right). \tag{8}$$

Rearranging completes the claim.

□

Remark. The non-negativity assumption is critical in the proof of Lemma 7.1. As a counter-example, consider the sets $C_1 = \{e1, e2\}, C_2 = \{e2, e3\}$ with $w(e1) = w(e3) = 1$ and $w(e2) = -1$. The union-bound fails for $w(C_1 \cup C_2)$.

Using Lemma 7.1, the proof of Thm. 2.1 goes as follows:

Proof. (Thm. 2.1) Multiplying the inequality obtained in the previous lemma by $\left(\frac{s-1}{s}\right)$ and adding to the inequality for step two, we get

$$w^*(C_1) + w^*(C_2) \geq \left(1 + \left(\frac{s-1}{s}\right)\right)\frac{OPT}{s}$$

We multiply this equation again by $\left(\frac{s-1}{s}\right)$ and add to the equation for step three, and so on. We get the following,

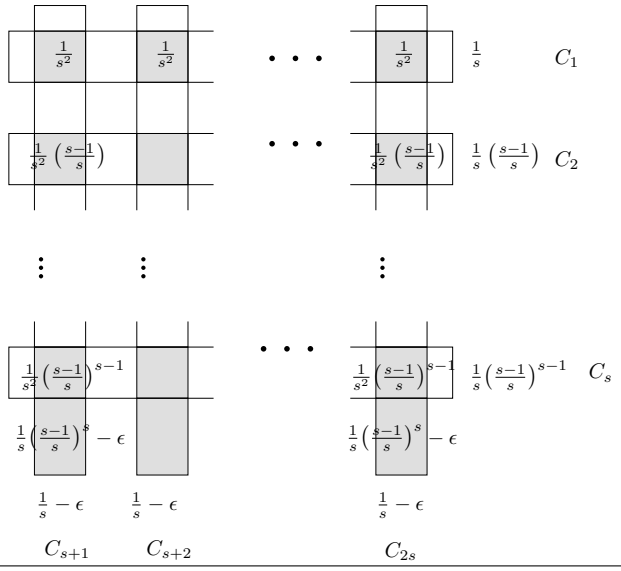
$$\sum_{j=1}^{k} w^*(C_j) \geq \left(1 - \left(\frac{s-1}{s}\right)^k\right) OPT$$

For $k = s$, we get,

$$\frac{\sum_{j=1}^{s} w^*(C_j)}{OPT} \geq \left(1 - \left(\frac{s-1}{s}\right)^s\right)$$

The left hand side is the ratio of the weight of the subset of $\mathcal{C}$ chosen by the greedy approach and the optimum solution i.e. that approximation factor and hence we have the above theorem. The fact that the above ratio is greater than $1 - \frac{1}{e}$ for all s is a trivial exercise. □

Figure 11 A tight example for the greedy strategy



7.1.2 Optimality

To prove Thm. 2.2, we construct tight examples for the greedy approach.

Proof. (Thm. 2.2) Fix a given s. We shall construct an example where the greedy approach can achieve an approximation ratio arbitrarily close to $1-(1-\frac{1}{s})^s$.

Let

$$\mathcal{A} = \{A_i\}_{i=1,\ldots,(s^2+s)}$$

$$\forall i,j \text{ s.t. } 0 \leq i < s, 1 \leq j \leq s,\ w(A_{i.s+j}) = \frac{1}{s^2}\left(\frac{s-1}{s}\right)^i$$

$$\forall j \text{ s.t. } 1 < j \leq s,\ w(A_{s^2+j}) = \frac{1}{s}\left(\frac{s-1}{s}\right)^s - \epsilon$$

The conformers are defined as follows

$$\mathcal{C} = \{C_i\}_{i=1,\ldots,2s}$$

$$\forall i \text{ s.t. } 1 \leq i \leq s,\ C_i = \bigcup_{j=(i-1).s+1}^{i.s} A_j$$

$$\forall i \text{ s.t. } \leq i \leq s,\ C_{s+i} = \bigcup_{j \equiv i \pmod{s}} A_j$$

Simple calculations lead us the following total weights for the conformers

$$\forall 1 \leq i \leq s,\ w(C_i) = \frac{1}{s}\left(\frac{s-1}{s}\right)^{i-1}$$

$$\forall 1 \leq i \leq s,\ w(C_{s+i}) = \frac{1}{s} - \epsilon$$

The optimum choice of $\mathcal{S}$ with $|\mathcal{S}| = s$ is clearly $\{C_i\}_{i=s+1,\ldots,2s}$ with total weight $1-s\epsilon$, whereas the greedy method would choose $\{C_i\}_{i=1,\ldots,s}$, with a maximum weight of $1-(1-\frac{1}{s})^s$, giving an approximation factor is arbitrarily close to $1-(1-\frac{1}{s})^s$. □

8 Appendix: Surface Decompositions

8.1 Approximating Factor for Problem 2

The following counter-example sets the approximation ratio for the greedy algorithm for the boundary surface case.

Proof. (Observation 1) Consider a large ball B, and place s small non-intersecting balls $(B_1, \ldots, B_s)$ with their centers on the surface of B. The surface of each B_i is now divided into 2 patches. To the patch which lies inside B, we assign a weight of s. To each surface patch of B covered by some B_i, we assign a weight of $1 + \epsilon$. All other surface patches are assigned a weight of 0.

The greedy strategy would first pick B because it has the largest exposed weight of $s(1 + \epsilon)$. Now picking any $s - 1$ of the B_i's would leave us with an exposed weight of only $s(1 + \epsilon) - (s - 1)(1 + \epsilon) = 1 + \epsilon$. On the opposite, a selection of the s smalls balls would have given us total exposed surface weight of s^2. This approximation factor arbitrarily close to $1/s^2$. □

8.2 Naive Algorithm for Surface Arrangement

Proof. (Thm. 3.1) To compute $w(G_{i-1} \cup \{C_j\})$, one needs the boundary of the corresponding balls. For a collection of i b balls contributed by i conformers, this is done in worst-case optimal time of $O(b^2 i^2)$, by first computing the regular triangulation of the balls, and then by retrieving the boundary of the union from the α-complex with $\alpha = 0$ [AE96]. The overall complexity is thus bounded by $\sum_{i=1}^{s}(n - i + 1)O(b^2 i^2)$, whence the claim. □

8.3 Priority-based Algorithm for Surface Arrangement

Notation. If X refers to a collection of conformers, $\cup X$ refers to the domain covered by these conformers, and $\partial \cup X$ refers to the boundary of the union of conformers in X. We shall abuse this notation, as we shall also use $\partial \cup X$ to refer to the finite number of spherical patches bounding the boundary of the union. Notice though, that the inclusion of a region r in the geometric boundary will be denoted $r \subset \cup X$, while the membership to the finite set describing this boundary will be denoted $r \in \partial \cup X$.

Computing the surface decompositions. Using the algorithm of [CL07, CL, CCCLT], we compute the arrangement on each sphere, induced by the intersection circles with other spheres. The output consists of:

– $D(S_i) = \{P_k\}$: patches on sphere S_i,

– $H(P_k)$: collection of spheres covering patch P_k,

from which we easily derive:

– $K(S_i)$: collection of patches covered by sphere S_i,

– $B(C_i)$: patches contributing to the boundary of conformer C_i.

Algorithm. We now present Algorithm 1, which is illustrated on Fig. 12.

Let G_{i-1} be the collection of conformers selected up to stage $i - 1$, and denote C_{s_i} the ith conformer selected. Also denote R_i the candidate conformers remaining once the ith conformer has been selected. In order to select C_{s_i}, we maintain a priority queue Q such that the key associated to a conformer C_l is $k(C_l) = w(G_{i-1} \cup \{C_l\}) - w(G_{i-1})$.

Apart from the heap itself, we shall use the following data structures:

– GB: greedy selection boundary, i.e. patches found on $\partial \cup G_{i-1}$,

– $H_Q(P_k)$: candidate conformers covering patch P_k.

As the arrangement calculation provides us with a list $H(P_k)$ of balls covering a given patch P_k, the list of conformers $H_Q(P_k)$ covering P_k is easily set up.

We shall also assume a patch found on the boundary of a candidate conformer has a status with respect to to G_{i-1}: $status(P_k) = covered$ iff $P_k \subset \cup G_{i-1}$, and *exposed* otherwise. Upon selection of conformer C_{s_i}, two types of patches have to be taken care of:

▷ **Case 1:** patches covered by C_{s_i}, which are found either on $\partial \cup G_{i-1}$ (Case 1a), or patches found on the boundary of conformers from R_i (Case 1b).

Consider sub-case 1a, i.e. a P_k patch found on $\partial \cup G_{i-1}$ which is covered by C_{s_i}. If this patch is also covered by another conformer C_l in R_i, the weight of this conformer has to be updated as $k(C_l) \leftarrow k(C_l) + w(P_k)$. Indeed, conformers C_{s_i} and C_l were competing in the queue, and both had been subtracted $w(P_k)$ to compare the relative increments $k(C_{s_i})$ and $k(C_l)$. Now that C_{s_i} has been selected, and since patch P_k has already been accounted for in the weight of conformer C_{s_i}, the weight of conformer C_l has to be corrected as indicated.

Consider now sub-case 1b, i.e. a patch P_k found on the boundary of a candidate conformer. This patch being now covered by C_{s_i}, it will not contribute to an increment of the boundary of the union, so that conformer C_l has to be updated as $k(C_l) \leftarrow k(C_l) - w(P_k)$.
▷ **Case 2:** patches found on the boundary $\partial \cup G_i$ contributed by conformer C_{s_i}. In selecting the $i+1$th conformer, such patches may get covered by candidate conformers. The weight of each such conformer C_l thus has to be updated as $k(C_l) \leftarrow k(C_l) - w(P_k)$. Note in passing that the fact that several candidates may cover such a patch is responsible for the afore-described sub-case 1a.

To prove Thm. 3.2, we shall assume that the priority queue is implemented using a Fibonacci heap, while dictionaries are handles using hash tables. Under these assumptions:

Proof. (Thm. 3.2) We first note that each hash-set operation and `UpdateKey` operation individually takes $O(1)$ amortized time, whereas the `RemoveMin` operation takes $O(\log n)$ time –using Fibonacci heaps.

The outermost loop and hence the `RemoveMin` operation is repeated s times. We now look at the calls of Update_H_lists. This function is called at most once for each conformer. The first loop in the function runs at most once for each primitive. For each primitive, the two inner most lines are executed as many times as there are patches that are covered by the primitive. Summing over all possible primitives and conformers, the number of times the inner most lines are executed is clearly bounded by τ. Now, consider the loop that repeats for all patches P_k that are covered by the primitive S_j. The statements inside the if loop are executed if the patch was on the boundary of the union of previously selected conformers. If this is the case, the patch no longer remains on the boundary after the execution of this part. So the lines inside the *if part* are executed at most once for each patch. In the *if part*, there is a loop that repeats for every candidate conformer that covers the patch. Summing over all possible patches, these lines are executed at most τ times. The *else part* takes constant time in each run, and is repeated for every candidate conformer that contains the patch. Thus, the else part is also repeated at most τ times. The second loop for the boundary patches repeats at most once for each patch. The inner loop there, again repeats for each candidate that contains the patch, hence the number of executions of the innermost statement is again bounded by τ.

Thus, overall the execution of the algorithm is bounded by $O(\tau + s \log n)$. □

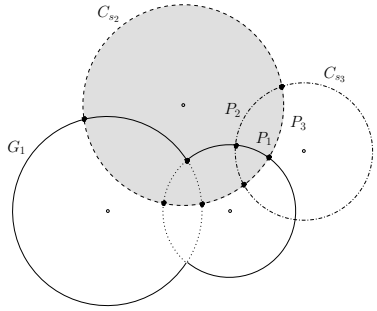
Algorithm 1 Greedy algorithm for surface decomposition.

```
W_t ← 0 /*Total weight returned*/
G_0 ← ∅/*Greedy Selection*/
GB ← ∅/*Greedy Selection Boundary*/
for i = 1 to s do
  RemoveMin: Pop C_{s_i} from queue
  G_i = G_{i-1} ∪ {C_{s_i}}
  Update_H_lists(C_{s_i})
  for all primitives S_j of C_{s_i} do
    /*Case 1: patches covered by S_j*/
    for all patches P_k ∈ K(S_j) /*covered by S_j*/ do
      /*Case 1a: patches on G_{i-1}*/
      if P_k ∈ GB /*P_k ∈ ∂ ∪ G_{i-1}*/ then
        GB ← GB\{P_k}
        for all C_l ∈ H_Q(P_k) /*candidates covering P_k*/ do
          UpdateKey: k(C_l) ← k(C_l) + w(P_k)
        /*Case 1b: patches of conformers in R_i*/
      else if status(P_k) = exposed /*P_k ⊄ ∪G_{i-1}*/ then
        Let C_l be the conformer patch P_k is on the boundary of
        UpdateKey: k(C_l) ← k(C_l) − w(P_k)
      status(P_k) ← covered
    /*Case 2: patches on the boundary of C_{s_i}*/
    for all P_k ∈ B(C_{s_i}) /* boundary of C_{s_i}*/ do
      if status(P_k) = exposed /*P_k ⊄ ∪G_{i-1}*/ then
        GB ← GB ∪ {P_k}
        for all C_l ∈ H_Q(P_k) /*candidates covering P_k*/ do
          UpdateKey: k(C_l) ← k(C_l) − w(P_k)
```

Algorithm 2 Algorithm $Update_H_lists(C_i)$

for all primitives S_j of C_i **do**
  for all patches $P_k \in K(S_j)$ /*covered by S_j*/ **do**
    if $C_i \in H_Q(P_k)$ **then**
      remove C_i from $H_Q(P_k)$

Figure 12 Greedy algorithm for surface weights. Patches triggering updates of keys upon selection of C_{s_2} are P_1, P_2, P_3.



9 Appendix: Material and Methods

9.1 `Direx` and `Loopy`

We selected two algorithms to generate loop conformers, which respectively yield dense and sparse ensembles of conformers. Algorithm `Direx` [SBL07], based on algorithm CONCOORD [dGvAS+97], handles a whole protein and processes all the atoms in the same way. The method performs perturbations of the atomic positions while preserving constraints on internal coordinates (bond lengths and dihedral angles). The generation of n conformers is iterative, since the kth conformer is taken as starting point for the generation of the $k+1$th one. Applying this algorithm to a loop from a PDB structure yields a collection of conformers spanning a relatively small region around the original loop in the PDB file.

Algorithm `Loopy` [XSH02] is a genetic algorithm that evolves a population of loops, the $k+1$th generation mixing a subset (survivors) of the kth generation together with new individuals derived from this subset. The main features of the algorithm are two-fold. First, the algorithm focuses on the backbone, onto which side-chains are added using a rotamer library. Second, the selection of the survivors uses a *colony* energy. This energy features a potential energy term, together with an entropy term encoding the spread of the neighborhood of a given conformation. This latter term accounts for the usual enthalpy-entropy competition, since a high internal energy conformation might be promoted thanks to a large entropy. This strategy naturally yields rather diverse sets of conformations.

Table 2 `Direx` versus `Loopy`: MSA for $n = 500$ conformers. The number of residues takes into account the two residues bounding all the conformers—these are common to all conformers.

PDBCODE	MSA `Direx`	MSA `Loopy`	Nb balls	Nb res.	RMSD
1BTH-atomic	1906.01	3423.34	108	12	5.7Å
1BTH-coarse	1639.72	3142.11	29	12	
1CGI-atomic	1867.17	3516.97	103	13	unres.
1CGI-coarse	1583.7	3032.78	28	13	
1OAZ-atomic	2535.6	4788.63	142	14	2.1Å
1OAZ-coarse	2262.51	4211.13	37	14	
3HHR-atomic	3835.2	15976.8	223	28	5.5Å
3HHR-coarse	3549.63	16345.7	67	28	

9.2 Geometric and Topological Assessment: Tables

Table 3 `Direx` data set. Comparison of the selection methods. See section 4.2 for notation.

PDB	$\frac{R_G}{R_H}$	$\frac{n_{H_{100\%}}}{n_G}$	$\frac{n_{H_{95\%}}}{n_G}$
1BTH-atomic	1.11	112.67	6.0
1CGI-atomic	1.12	154.33	1.67
1OAZ-atomic	1.13	1.67	1.33
3HHR-atomic	1.09	9.67	2.67
1BTH-coarse	1.14	67.2	3.6
1CGI-coarse	1.12	160.0	2.33
1OAZ-coarse	1.15	1.67	1.33
3HHR-coarse	1.11	10.67	2.67

Table 4 `Loopy` data set. Comparison of the selection methods. See section 4.2 for notation.

PDB	$\frac{R_G}{R_H}$	$\frac{n_{H_{100\%}}}{n_G}$	$\frac{n_{H_{95\%}}}{n_G}$
1BTH-atomic	1.18	9.5	7.0
1CGI-atomic	1.14	79.5	28.17
1OAZ-atomic	1.21	42.64	10.82
3HHR-atomic	1.54	4.3	3.8
1BTH-coarse	1.25	10.86	8.29
1CGI-coarse	1.34	79.5	11.0
1OAZ-coarse	1.27	44.67	10.89
3HHR-coarse	1.56	13.31	3.0

Table 5 `Direx` data set. Comparing the evolution of the first Betti number up to $n_{G_{100\%}}$ conformers selected; Left: `Greedy`; Right: `HClust`. m, M, μ and med respectively stand for $\min, \max, mean, median$.

PDB	$n_{G_{100\%}}$	$m(\beta_1)$	$M(\beta_1)$	$\mu(\beta_1)$	$med(\beta_1)$	$m(\beta_1)$	$M(\beta_1)$	$\mu(\beta_1)$	$med(\beta_1)$
1BTH-atomic	28	7	16	9.38	8	0	7	0.44	0
1CGI-atomic	10	2	12	8.06	8	0	8	0.45	0
1OAZ-atomic	13	5	21	7.07	6	0	19	0.44	0
3HHR-atomic	6	6	30	13.95	11.5	0	31	1.05	0
1BTH-coarse	17	0	12	9.63	11	0	11	0.96	1
1CGI-coarse	22	0	17	6.49	7	0	8	1.14	1
1OAZ-coarse	14	1	18	9.79	10	0	14	1.53	2
3HHR-coarse	11	4	36	16.16	15	1	23	2.22	1

Table 6 `Loopy` data set. Comparing the evolution of the first Betti number up to $n_{G_{100\%}}$ conformers selected; Left: `Greedy`; Right: `HClust`. m, M, μ and med respectively stand for $\min, \max, mean, median$.

PDB	$n_{G_{100\%}}$	$m(\beta_1)$	$M(\beta_1)$	$\mu(\beta_1)$	$med(\beta_1)$	$m(\beta_1)$	$M(\beta_1)$	$\mu(\beta_1)$	$med(\beta_1)$
1BTH-atomic	39	3	44	25.08	24	1	20	3.78	2
1CGI-atomic	16	4	38	23.68	22	1	27	2.6	2
1OAZ-atomic	37	4	43	25.82	24	4	18	7.58	5
3HHR-atomic	36	17	342	283.12	306	26	163	86.56	74
1BTH-coarse	32	0	51	36.26	35	2	25	6.73	6
1CGI-coarse	21	0	59	30.15	26	0	30	5.67	5
1OAZ-coarse	28	0	77	61.67	63	7	35	19.99	19
3HHR-coarse	46	1	492	382.59	452	6	219	149.02	141

Table 7 Selection by `HClust` versus selection by `Greedy` for coarse models: C_α-rmsd of loops selected with respect to the native bound loop; Min. and Max. stand for the minimum and the maximum of these values. $\delta = Max - Min$, the subscript G or H standing for the algorithm used, `Greedy` or `HClust`.

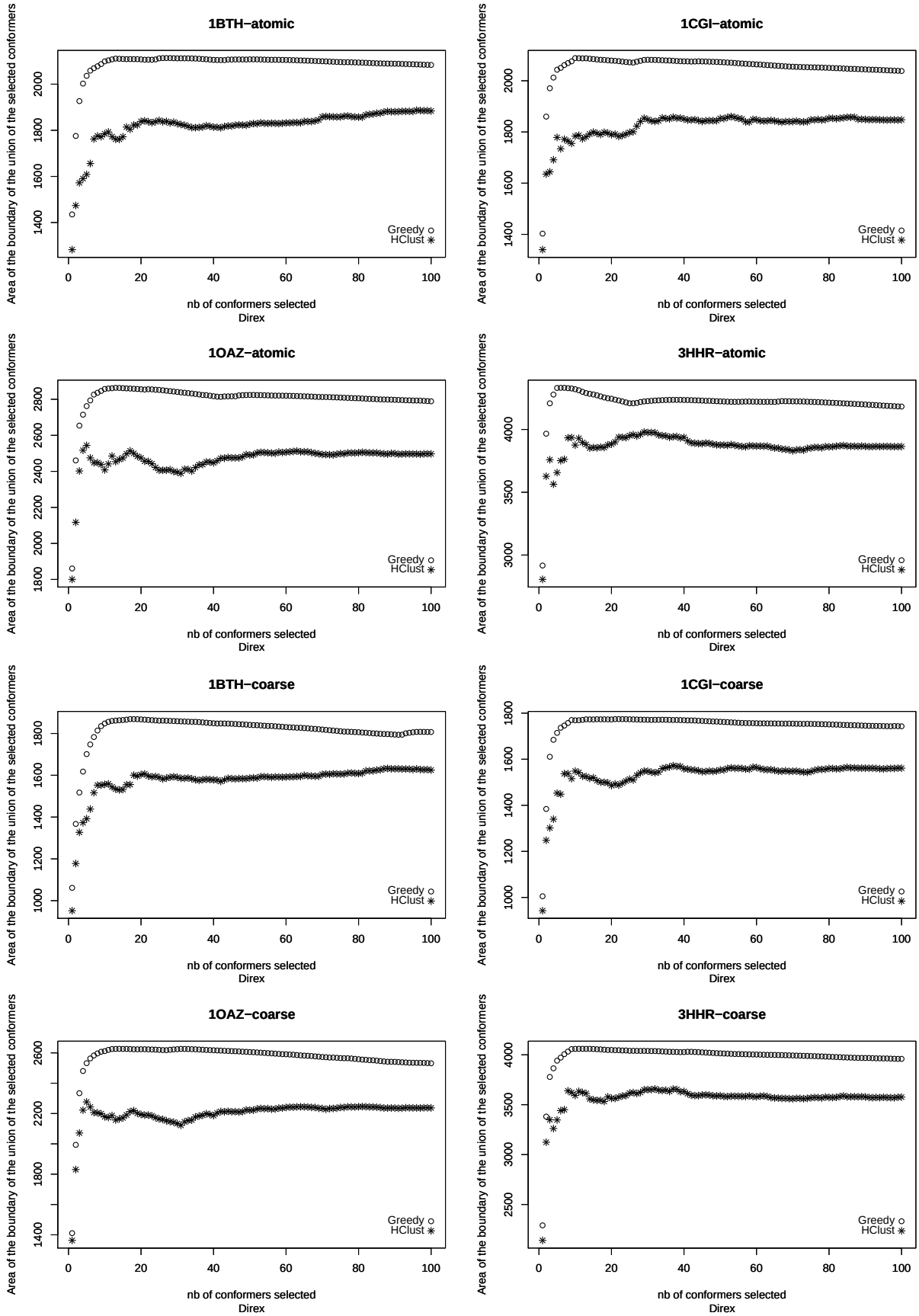
File	Loop generated by	Selected by	Selection size	Min.	Max.	δ	$\frac{\delta_G - \delta_H}{\delta_H}$
1BTH	`Loopy`	Greedy	10	2.767	10.15	7.383	0.392
1BTH	`Loopy`	HClust	10	2.693	7.996	5.303	
1BTH	`Loopy`	Greedy	15	2.767	10.15	7.383	0.315
1BTH	`Loopy`	HClust	15	2.693	8.307	5.614	
1BTH	`Loopy`	Greedy	30	2.581	10.15	7.569	-0.052
1BTH	`Loopy`	HClust	30	2.162	10.15	7.988	
1BTH	`Direx`	Greedy	10	0.5627	3.501	2.9383	0.184
1BTH	`Direx`	HClust	10	0.9152	3.396	2.4808	
1BTH	`Direx`	Greedy	15	0.5627	3.501	2.9383	0.184
1BTH	`Direx`	HClust	15	0.9152	3.396	2.4808	
1BTH	`Direx`	Greedy	30	0.5627	3.679	3.1163	0.203
1BTH	`Direx`	HClust	30	0.8562	3.446	2.5898	
1CGI	`Loopy`	Greedy	10	3.614	10.4	6.786	0.27
1CGI	`Loopy`	HClust	10	3.413	8.755	5.342	
1CGI	`Loopy`	Greedy	15	3.614	10.91	7.296	0.326
1CGI	`Loopy`	HClust	15	3.413	8.916	5.503	
1CGI	`Loopy`	Greedy	30	3.614	10.95	7.336	-0.078
1CGI	`Loopy`	HClust	30	2.441	10.4	7.959	
1CGI	`Direx`	Greedy	10	0.7777	4.442	3.6643	0.637
1CGI	`Direx`	HClust	10	1.364	3.602	2.238	
1CGI	`Direx`	Greedy	15	0.7777	4.442	3.6643	0.336
1CGI	`Direx`	HClust	15	1.364	4.106	2.742	
1CGI	`Direx`	Greedy	30	0.7777	4.442	3.6643	0.336
1CGI	`Direx`	HClust	30	1.364	4.106	2.742	
1OAZ	`Loopy`	Greedy	10	4.237	17.32	13.083	0.58
1OAZ	`Loopy`	HClust	10	3.327	11.61	8.283	
1OAZ	`Loopy`	Greedy	15	4.237	17.35	13.113	0.392
1OAZ	`Loopy`	HClust	15	3.079	12.5	9.421	
1OAZ	`Loopy`	Greedy	30	2.716	17.35	14.634	0.68
1OAZ	`Loopy`	HClust	30	3.079	11.79	8.711	
1OAZ	`Direx`	Greedy	10	0.8034	6.967	6.1636	0.353
1OAZ	`Direx`	HClust	10	1.933	6.49	4.557	
1OAZ	`Direx`	Greedy	15	0.8034	6.967	6.1636	0.219
1OAZ	`Direx`	HClust	15	1.111	6.169	5.058	
1OAZ	`Direx`	Greedy	30	0.8034	6.967	6.1636	0.221
1OAZ	`Direx`	HClust	30	1.111	6.159	5.048	
3HHR	`Loopy`	Greedy	10	8.517	21.16	12.643	-0.288
3HHR	`Loopy`	HClust	10	5.521	23.27	17.749	
3HHR	`Loopy`	Greedy	15	8.517	23.27	14.753	-0.169
3HHR	`Loopy`	HClust	15	5.521	23.27	17.749	
3HHR	`Loopy`	Greedy	30	8.517	23.27	14.753	-0.169
3HHR	`Loopy`	HClust	30	5.521	23.27	17.749	
3HHR	`Direx`	Greedy	10	0.4793	5.389	4.9097	0.092
3HHR	`Direx`	HClust	10	0.8399	5.335	4.4951	
3HHR	`Direx`	Greedy	15	0.4793	5.485	5.0057	0.114
3HHR	`Direx`	HClust	15	0.8399	5.335	4.4951	
3HHR	`Direx`	Greedy	30	0.4793	5.485	5.0057	0.111
3HHR	`Direx`	HClust	30	0.8399	5.346	4.5061	

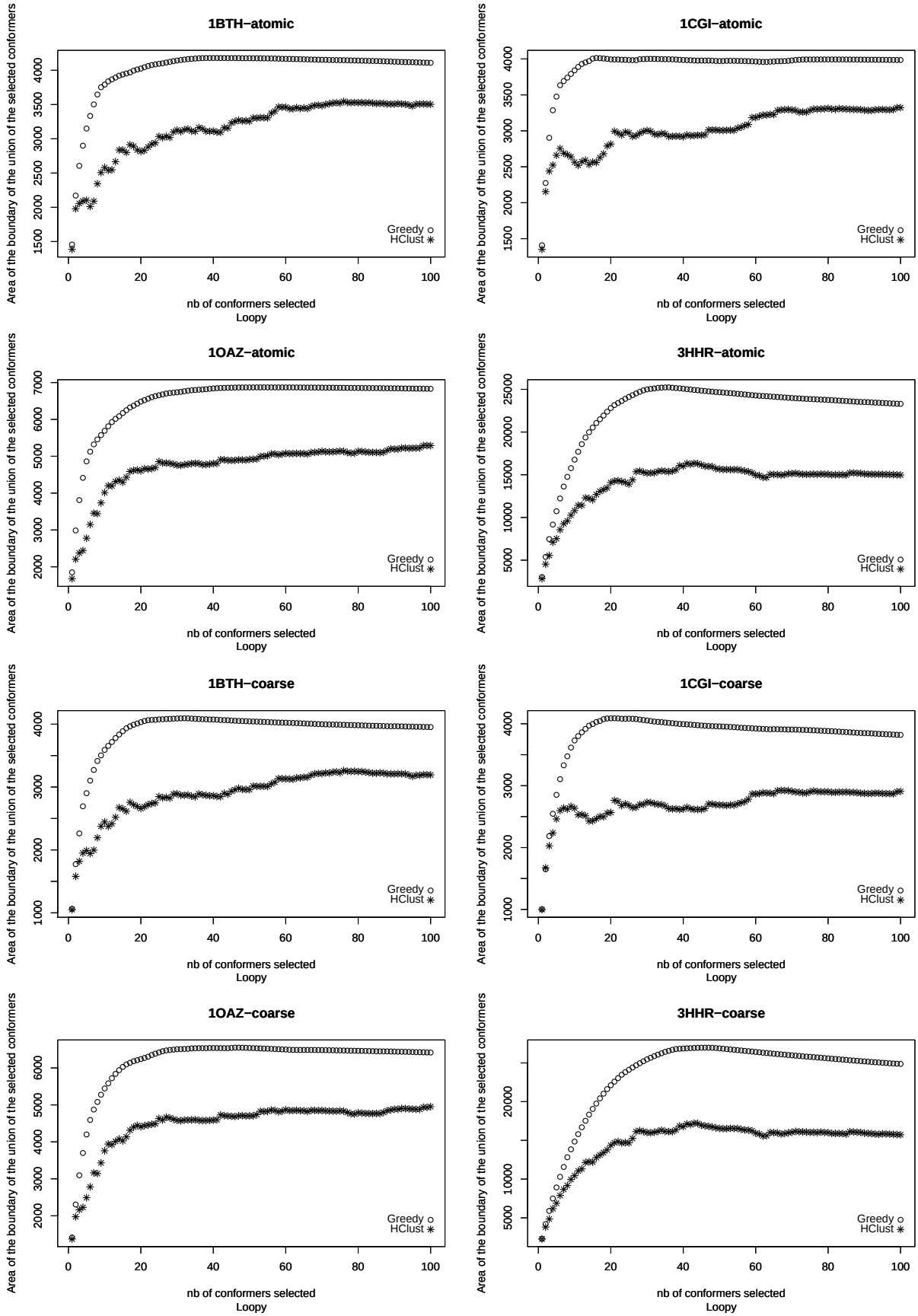
Table 8 Selection by `HClust` versus selection by `Greedy` for atomic models: C_α-rmsd of loops selected with respect to the native bound loop; Min. and Max. stand for the minimum and the maximum of these values. $\delta = Max - Min$, the subscript G or H standing for the algorithm used, `Greedy` or `HClust`.

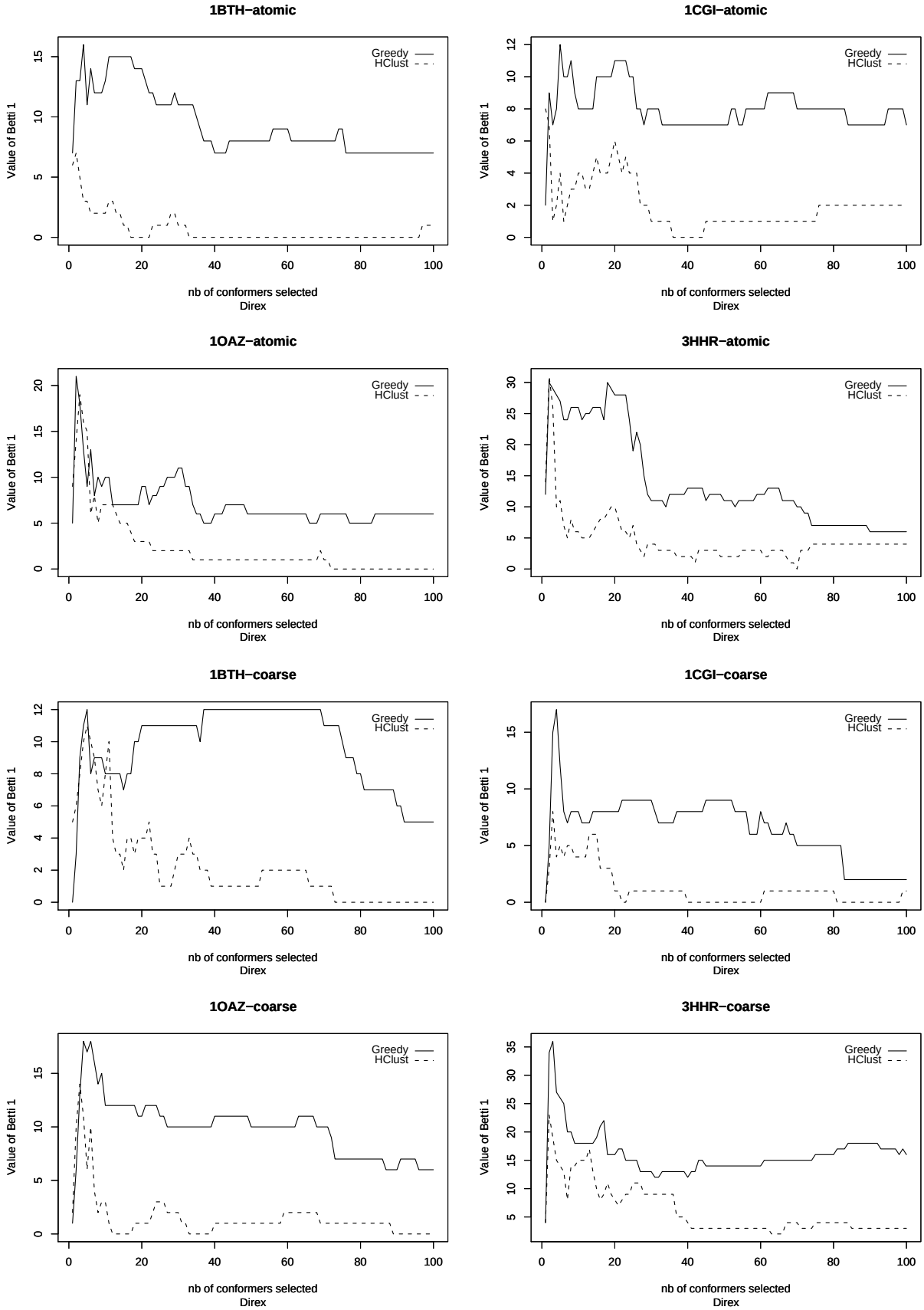
File	Loop generated by	Selected by	Selection size	Min.	Max.	δ	$\frac{\delta_G - \delta_H}{\delta_H}$
1BTH	`Loopy`	Greedy	10	2.693	10.15	7.457	0.406
1BTH	`Loopy`	HClust	10	2.693	7.996	5.303	
1BTH	`Loopy`	Greedy	15	2.581	10.15	7.569	0.348
1BTH	`Loopy`	HClust	15	2.693	8.307	5.614	
1BTH	`Loopy`	Greedy	30	2.581	10.15	7.569	-0.052
1BTH	`Loopy`	HClust	30	2.162	10.15	7.988	
1BTH	`Direx`	Greedy	10	0.9006	3.528	2.6274	0.059
1BTH	`Direx`	HClust	10	0.9152	3.396	2.4808	
1BTH	`Direx`	Greedy	15	0.9006	3.679	2.7784	0.12
1BTH	`Direx`	HClust	15	0.9152	3.396	2.4808	
1BTH	`Direx`	Greedy	30	0.9006	3.679	2.7784	0.073
1BTH	`Direx`	HClust	30	0.8562	3.446	2.5898	
1CGI	`Loopy`	Greedy	10	5.934	10.55	4.616	-0.136
1CGI	`Loopy`	HClust	10	3.413	8.755	5.342	
1CGI	`Loopy`	Greedy	15	3.614	10.91	7.296	0.326
1CGI	`Loopy`	HClust	15	3.413	8.916	5.503	
1CGI	`Loopy`	Greedy	30	3.614	10.95	7.336	-0.078
1CGI	`Loopy`	HClust	30	2.441	10.4	7.959	
1CGI	`Direx`	Greedy	10	1.45	4.442	2.992	0.337
1CGI	`Direx`	HClust	10	1.364	3.602	2.238	
1CGI	`Direx`	Greedy	15	1.45	4.442	2.992	0.091
1CGI	`Direx`	HClust	15	1.364	4.106	2.742	
1CGI	`Direx`	Greedy	30	1.437	4.442	3.005	0.096
1CGI	`Direx`	HClust	30	1.364	4.106	2.742	
1OAZ	`Loopy`	Greedy	10	2.716	17.35	14.634	0.767
1OAZ	`Loopy`	HClust	10	3.327	11.61	8.283	
1OAZ	`Loopy`	Greedy	15	2.716	17.35	14.634	0.553
1OAZ	`Loopy`	HClust	15	3.079	12.5	9.421	
1OAZ	`Loopy`	Greedy	30	2.716	17.35	14.634	0.68
1OAZ	`Loopy`	HClust	30	3.079	11.79	8.711	
1OAZ	`Direx`	Greedy	10	0.8034	6.71	5.9066	0.296
1OAZ	`Direx`	HClust	10	1.933	6.49	4.557	
1OAZ	`Direx`	Greedy	15	0.8034	6.71	5.9066	0.168
1OAZ	`Direx`	HClust	15	1.111	6.169	5.058	
1OAZ	`Direx`	Greedy	30	0.8034	6.71	5.9066	0.17
1OAZ	`Direx`	HClust	30	1.111	6.159	5.048	
3HHR	`Loopy`	Greedy	10	11.11	23.27	12.16	-0.315
3HHR	`Loopy`	HClust	10	5.521	23.27	17.749	
3HHR	`Loopy`	Greedy	15	11.11	23.27	12.16	-0.315
3HHR	`Loopy`	HClust	15	5.521	23.27	17.749	
3HHR	`Loopy`	Greedy	30	8.611	23.27	14.659	-0.174
3HHR	`Loopy`	HClust	30	5.521	23.27	17.749	
3HHR	`Direx`	Greedy	10	1.37	5.344	3.974	-0.116
3HHR	`Direx`	HClust	10	0.8399	5.335	4.4951	
3HHR	`Direx`	Greedy	15	1.37	5.344	3.974	-0.116
3HHR	`Direx`	HClust	15	0.8399	5.335	4.4951	
3HHR	`Direx`	Greedy	30	1.37	5.344	3.974	-0.118
3HHR	`Direx`	HClust	30	0.8399	5.346	4.5061	

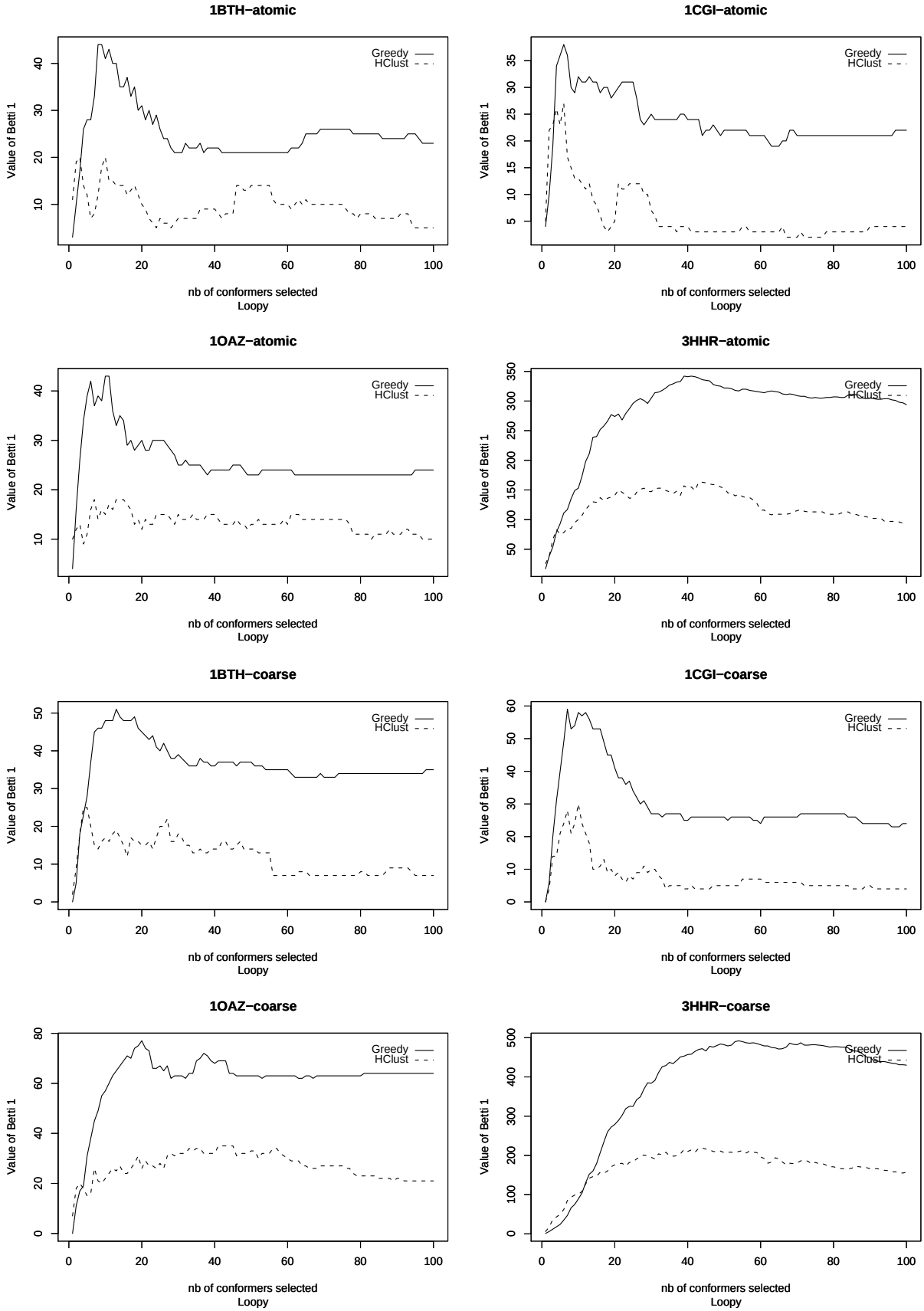
9.3 Geometric and Topological Assessment: Graphs

This section presents the plots of the variation of the MSA and of the Betti number β_1, as a function of the selection size. For each statistic, each model features four plots: {`Direx`, `Loopy`} × {atomic resolution, coarse grain}. Results are discussed in section 4.2.


1BTH−atomic
1CGl−atomic
1OAZ−atomic
3HHR−atomic
1BTH−coarse
1CGl−coarse
1OAZ−coarse
3HHR−coarse
Area of the boundary of the union of the selected conformers
nb of conformers selected
Direx
Greedy
HClust


1BTH−atomic
1CGI−atomic
1OAZ−atomic
3HHR−atomic
1BTH−coarse
1CGI−coarse
1OAZ−coarse
3HHR−coarse
Area of the boundary of the union of the selected conformers
nb of conformers selected
Loopy
Greedy
HClust


1BTH−atomic
1CGI−atomic
1OAZ−atomic
3HHR−atomic
1BTH−coarse
1CGI−coarse
1OAZ−coarse
3HHR−coarse
Greedy
HClust
Value of Betti 1
nb of conformers selected
Direx


1BTH−atomic
1CGI−atomic
1OAZ−atomic
3HHR−atomic
1BTH−coarse
1CGI−coarse
1OAZ−coarse
3HHR−coarse
Greedy
HClust
Value of Betti 1
nb of conformers selected
Loopy

9.4 Graphs: Docking Assessment

For a given system, the 6 plots are organized as follows: the first two correspond to the sanity check B/B/B-1 and U/B/B-1; the next (respectively last) two, namely B/B/B-`HClust`-10 and B/B/B-`Greedy`-10 (respectively U/B/B-`HClust`-10, U/B/B-`Greedy`-10) allow the comparison of `Greedy` and `HClust` for the Bound (respectively Unbound) version of the receptor. Recall that flexible regions on `1BTH`, `1CGI` and `1OAZ` feature 10, 11 and 12 amino acids respectively on their receptor. These graphs are discussed in section 5.4.

Figure 13 1BTH: binning the docking tests using the Bound and Unbound forms of the receptor . See text for details.

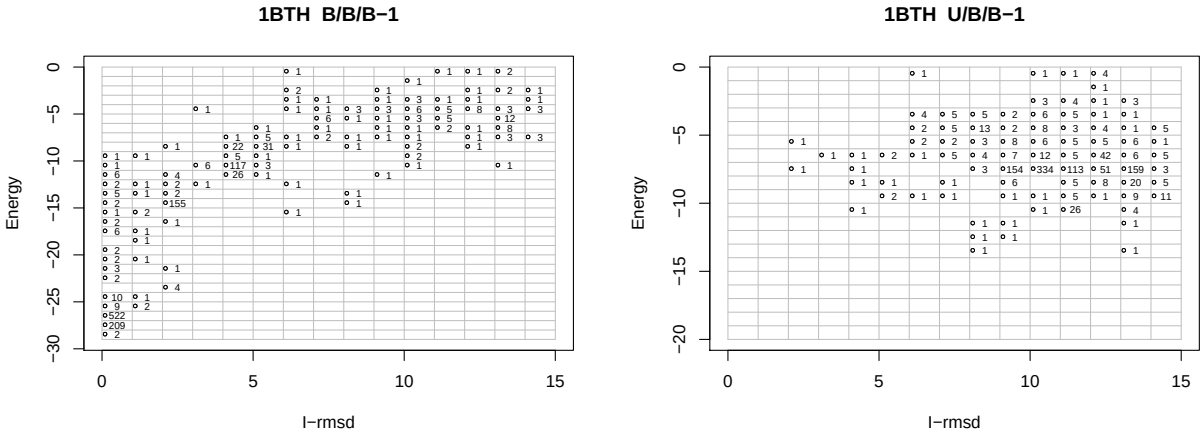


Figure 14 1BTH: binning the docking tests using the Bound form of the receptor. See text for details.

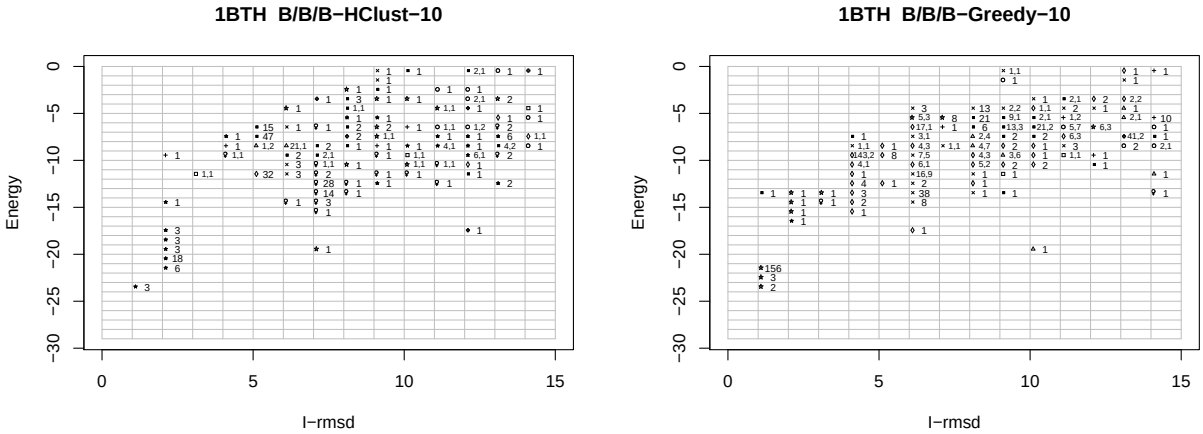


Figure 15 1BTH: binning the docking tests using the Unbound form of the receptor. See text for details.

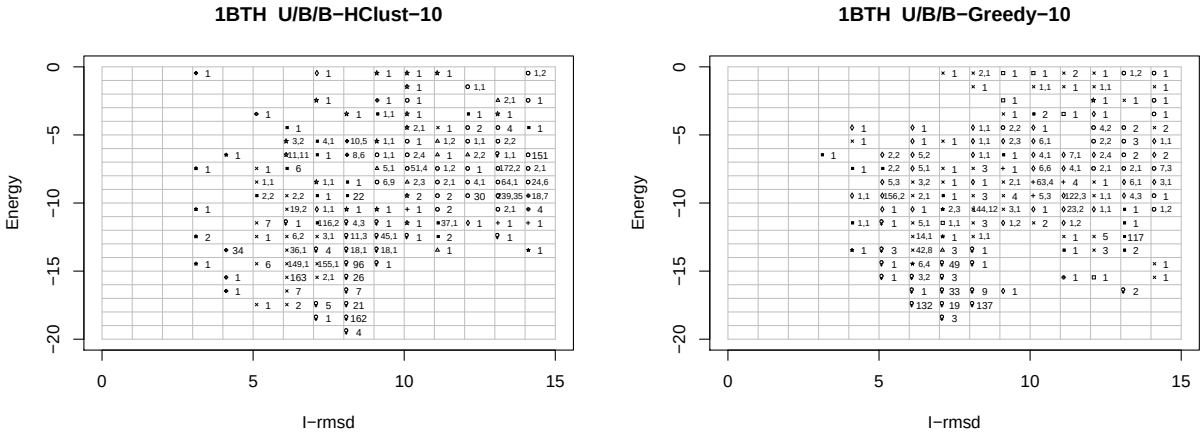


Figure 16 `1CGI`: binning the docking tests using the Bound and Unbound forms of the receptor . See text for details.

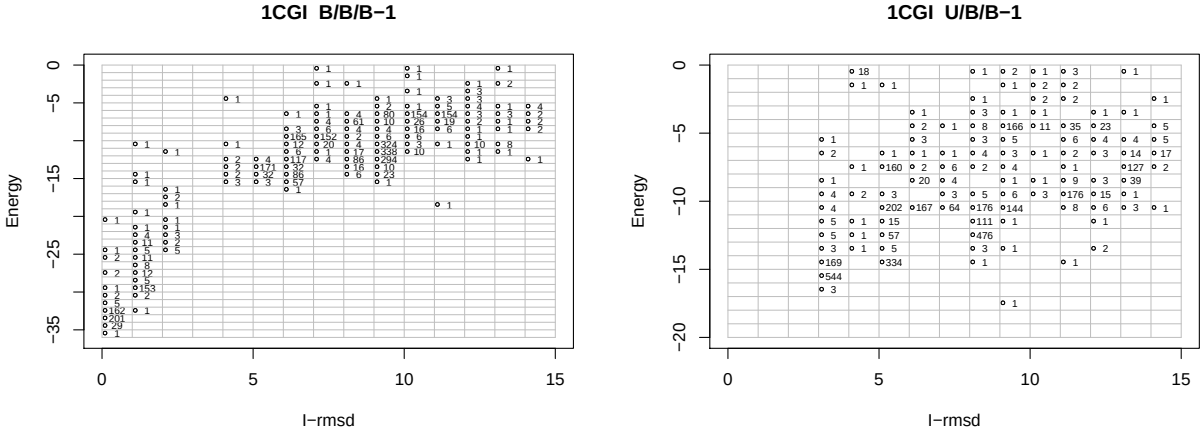


Figure 17 `1CGI`: binning the docking tests using the Bound form of the receptor. See text for details.

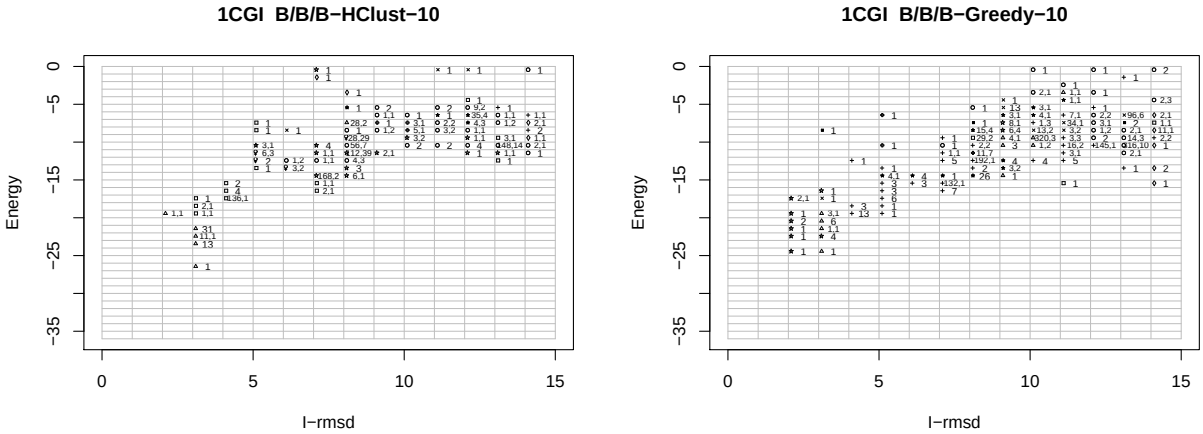


Figure 18 `1CGI`: binning the docking tests using the Unbound form of the receptor. See text for details.

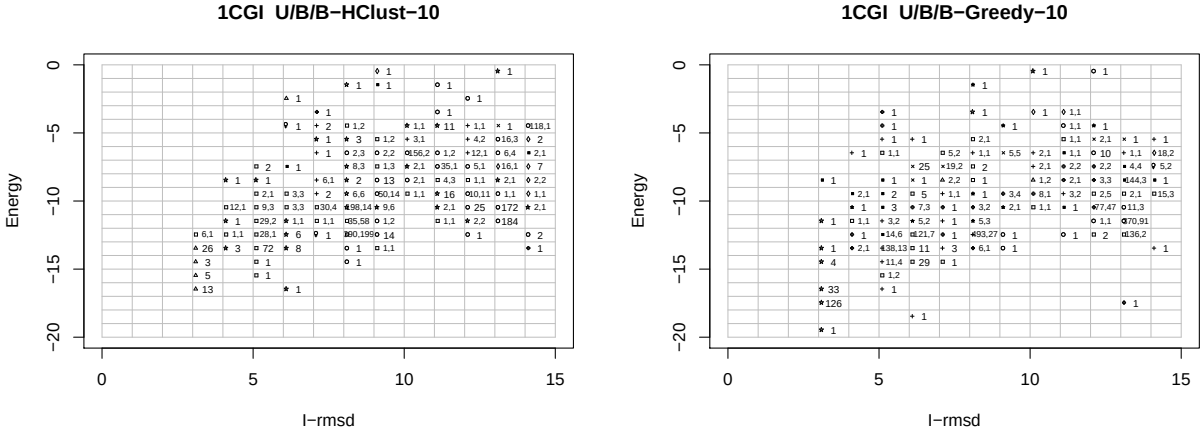


Figure 19 `1OAZ`: binning the docking tests using the Bound and Unbound forms of the receptor . See text for details.

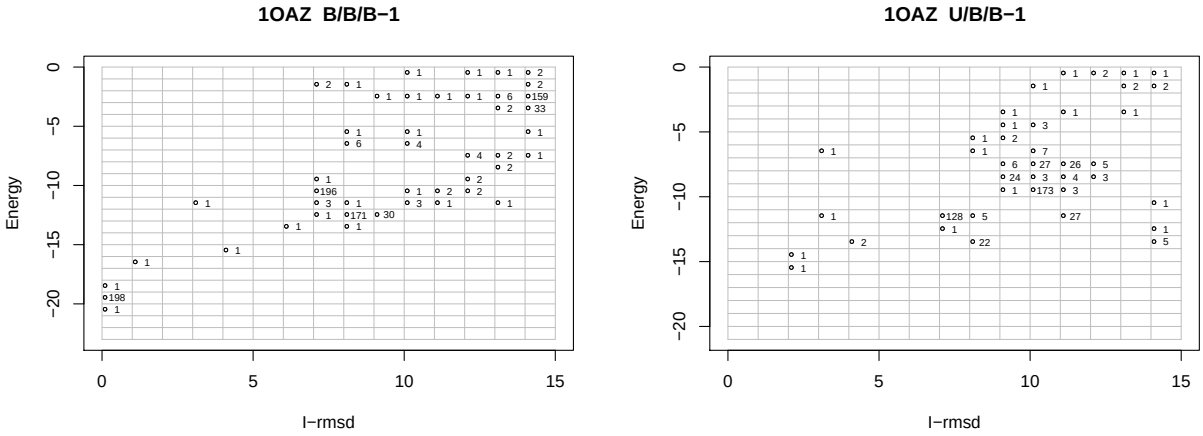


Figure 20 `1OAZ`: binning the docking tests using the Bound form of the receptor. See text for details.

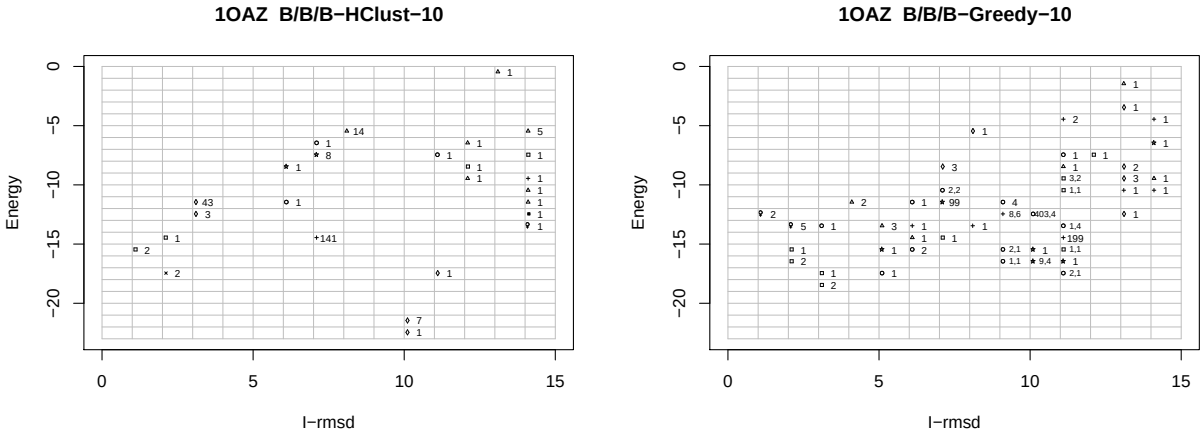
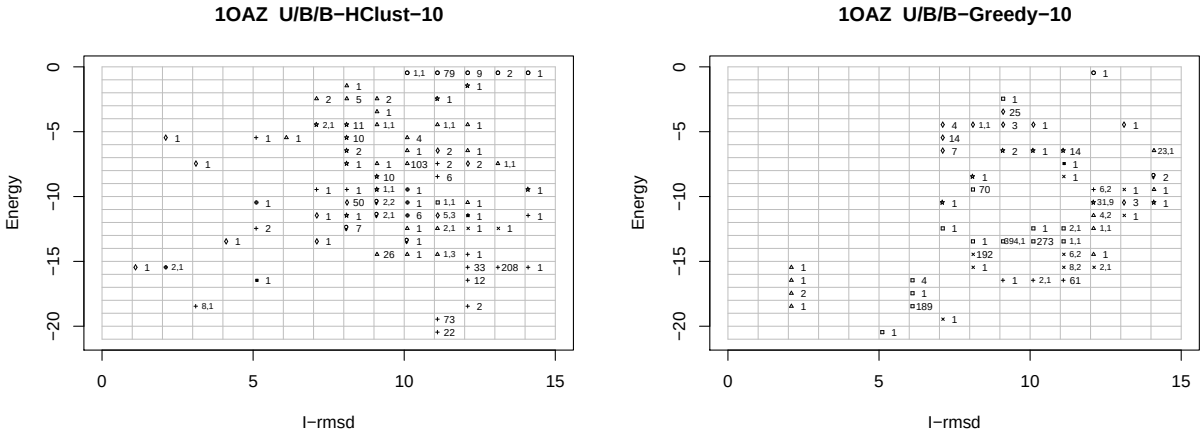


Figure 21 `1OAZ`: binning the docking tests using the Unbound form of the receptor. See text for details.



Contents

1 Introduction **3**
1.1 On the Importance of Diverse Conformational Ensembles . . . 3
1.2 Contributions and Paper Overview . . . 3

2 Selecting Conformers: the Combinatorial Viewpoint **6**
2.1 Arrangements of Balls and Spheres: Volume and Surface Decompositions . . . 6
2.2 Optimization Problems . . . 6
2.3 Instantiations to Conformer Selection . . . 7
2.4 The Greedy Strategy . . . 7
2.4.1 The Strategy and its Guarantees . . . 7
2.4.2 Application to Conformer Selections . . . 8

3 Material and Methods **8**
3.1 Data Sets and Conformer Generation Methods . . . 8
3.2 Greedy Selection: Implementation . . . 9
3.3 Conformer Selection Methods . . . 10

4 Diverse Ensembles: Geometric and Topological Assessment **10**
4.1 Statistics of Interest: Geometry vs. Topology . . . 10
4.2 Results . . . 11

5 Diverse Ensembles: Docking Assessment **14**
5.1 Docking as a Multi-scale Process . . . 14
5.2 Docking Protocols . . . 14
5.3 Initial Conditions for a Protocol . . . 15
5.4 Results . . . 16
5.5 Running Times . . . 16

6 Conclusion **18**

7 Appendix: Volumetric Decompositions **21**
7.1 Greedy: Approximation Factor and Optimality . . . 21
7.1.1 Approximation Factor . . . 21
7.1.2 Optimality . . . 22

8 Appendix: Surface Decompositions **23**
8.1 Approximating Factor for Problem 2 . . . 23
8.2 Naive Algorithm for Surface Arrangement . . . 23
8.3 Priority-based Algorithm for Surface Arrangement . . . 23

9 Appendix: Material and Methods **25**
9.1 `Direx` and `Loopy` . . . 25
9.2 Geometric and Topological Assessment: Tables . . . 26
9.3 Geometric and Topological Assessment: Graphs . . . 30
9.4 Graphs: Docking Assessment . . . 35

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

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

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