On the Characterization and Selection of Diverse Conformational Ensembles

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

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On the Characterization and Selection of Diverse Conformational Ensembles

S. Loriot — S. Sachdeva — K. Bastard — C. Prévost — F. Cazals

N° 6503

Février 2008

Thème BIO

Rapport de recherche

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

On the Characterization and Selection of Diverse Conformational Ensembles

S. Loriot [*] , S. Sachdeva [†] , K. Bastard [‡] , C. Prévost [§] , F. Cazals [¶]

Thème BIO — Systèmes biologiques
Projet ABS

Rapport de recherche n° 6503 — Février 2008 — 35 pages

Abstract: As of today, a number of flexible docking experiments fail due to the lack of candidate conformations close from the bound ones. To tackle these difficult cases, which usually feature large re-arrangements, the route of choice consists of pre-generating discrete ensembles of conformers. But for a flexible fragment (rotamer, loop, domain), the fact that the generation of such ensembles does not take into account the whole environment of the fragment (in the whole protein or complex), prevents, in general, to directly relate the conformer energy 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, so as 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. This paper makes three contributions in this perspective.

First, we formulate several conformer selection problems as a geometric optimization problems, so as to maximize spatial coverage in several guises. We propose to solve these problems using greedy algorithms, for which approximation bounds are given. Second, we develop several algorithms for one specific problem, that of optimizing the Molecular Surface Area (MSA) of the selection. Third, we experiment on the loops of four proteins (1BTH, 1CGI, 1OAZ, 3HHR), for dense and sparse conformational ensembles. Third, by focusing on the MSA and the Betti numbers of the union 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.

[*] INRIA Sophia-Antipolis, Algorithms-Biology-Structure, 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

We therefore expect our characterization and selection of diverse conformational ensembles to improve the processing of challenging docking experiments.

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

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

Résumé : Nombre de simulations de docking échouent dans le cas ou les partenaires présentent des degrés de flexibilité importants. Pour faire face à ces cas, la stratégie qui prévaut consiste à pré-générer un ensemble discret de conformations ou *conformères*. Mais pour une région flexible donnée (rotamère, boucle, domaine), le fait que la génération de ces ensembles ne prennent pas en compte l'environnement de la région dans la protéine ou le complexe, empêche, en général, d'établir un lien direct entre l'énergie du conformère et l'équilibre thermodynamique entre les conformations. Cette observation milite pour le développement de méthodes générant une couverture plutôt uniforme de l'espace des conformations, de façon à garder tous les conformères ne présentant pas des contre-indication stérique évidente. Ce voeux pieux se heurte toutefois à une difficulté majeure: caractériser la diversité conformationnelle, de façon à maximiser la diversité des conformères. Ce travail présente trois contributions dans cette perspective.

Tout d'abord, nous formulons plusieurs problèmes de sélection de conformères comme des problèmes d'optimisation géométrique, de façon à maximiser la couverture spatiale. Nous proposons de résoudre ces problèmes par l'algorithme glouton, pour lequel des facteurs d'approximation sont prouvés. Nous développons ensuite deux algorithmes pour un problème donne, à savoir la maximisation de l'aire de la surface moléculaire des conformères (MSA) sélectionnés. Enfin, nous présentons des résultats expérimentaux pour les boucles de quatre protéines (1BTH, 1CGI, 1OAZ, 3HHR), et ce pour des ensembles de conformères denses et épars. En étudiant la variation de la MSA et des nombres de Betti, nous montrons que notre stratégie génère une MSA équivalente à celle d'algorithmes classiques de clustering, mais en utilisant un nombre de conformères plus faible de un à deux ordres de grandeur.

De par la diversité des ensembles sélectionnés, ce travail devrait contribuer au traitement de cas difficiles de docking flexible.

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

1 Introduction

1.1 Flexibility in Docking Methods

Ensembles in molecular modeling. Protein-protein interactions are paramount to all biological processes, but their prediction from unbound geometries faces major difficulties, as evidenced by the medium to incorrect predictions carried out on flexible systems within the CAPRI experiment [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 (interaction) 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 (protein), local (side chain), or intermediate (loops or domains) scales. At the global level, conformations of a whole partner may be generated. For example, it is observed in [GLN04] that cross-docking conformers (sequentially docking each possible pair of conformers) increases the chance of finding the correct bound geometry. At the local level, libraries of rotamers can be used, for example when positioning the side chains in the last step of homology modeling —a process which consists of predicting the $3D$ structure of a protein based on the structure of a known homolog(s). To optimize the internal energy of the modeled protein, one needs to find *coherent* rotamers. This problem can be tackled by considering the rotamers sequentially [CSD03], or by dealing ensembles of weighted rotamers [KD94]. This latter approach relies on the mean field theory, and consists of iteratively modifying weights associated to the rotamers, until the best combination is found. Finally, at the intermediate level, if a protein features flexible parts, such parts can be advantageously considered as collections of conformations each time they interact with the rest of the structure or with interaction partners. In the field of small ligand docking, the program FlexE assembles composite proteins from conformer ensembles of their flexible parts, before performing cross-docking simulations [CBRL01]. For homology modeling, mean-field theory has been used to determine the optimal conformation of protein fragments within the protein frame [KD95]. Recently, mean-field theory has also been applied to conformation ensembles of protein loops or domains for macromolecular docking [BTLP03, BPZ06].

Generating and selecting conformers: energy versus geometry. Representing flexibility through an ensemble of conformers is computationally feasible only if this number is not too important. It is therefore essential for this reduced number of conformers to be as much representative as possible of the conformational space available to the flexible molecule or molecule fragment. More generally, conformers being of interest for a number of applications, which criteria (geometric or energetic) should one use to generate and/or select them? In a statistical viewpoint, energy should be the criteria 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. To keep calculations tractable, methods undertaking this task favor geometric calculations, and defer energy calculations to later stages [CSdA$^+$05, DYM$^+$07]. 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 problems 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 in different environments. 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 uniformly sample the space available to the flexible element.

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, as the balls may represent atoms, or may model residues. In the first case, we shall be using balls with Van der Waals radii, so that the molecular surface defined is the Van der Waals surface. In the second one, we shall trim an atomic model down to a model representing residues. To measure the dissimilarity of two conformers, we compute the RMSD of their balls, i.e. of the atomic balls for the atomic model, and of the balls representing the residues for the coarse model.

Problem addressed. As argued in section 1.1, sampling uniformly the conformational space available is an important requirement. We actually wish to solve the following problem:

Given a pre-computed collection of n conformers and an integer $s < n$, report a *selection* of s conformers that *best represents* the n conformers according to some geometric 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/decomposition*. Similarly, the decomposition of each sphere by the intersection circles with other spheres defines a *(surface) arrangement/decomposition*. See Figs. 1 and 4 for a $2D$ illustration. Using these arrangements, we investigate several geometric optimization problems whose output is the selection. These problems aim at maximizing the *spatial occupancy* of the selection, in several guises. For example, we may wish to report the s conformers maximizing (i) the volume occupied by these conformers (ii) the molecular surface area (MSA) of the union of the conformers –see Fig. 2.

Paper overview. A number of 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 focus on one such problem, namely that of reporting a selection maximizing the MSA area of the union of the conformers. Results on conformers corresponding to four proteins loops featuring from 11 to 27 residues are reported in section 4, and discussed in section 5.

Figure 1 Example $2D$ conformers, each consisting of four balls –1st and 4th balls are common. The (two dimensional) volume occupied by the two conformers is decomposed into 19 cells (squared numerals). The circled numerals feature the surface arrangement of the ball centered at a_1, based on intersections with neighboring balls.

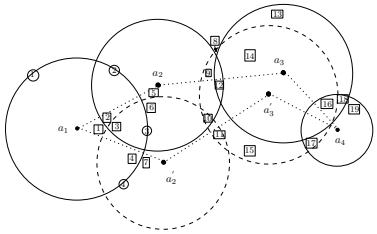


Figure 2 Selecting conformers of a flexible loop; here the endpoints of all conformers in each panel coincide. Compare the 5 interpenetrating conformers returned by standard clustering algorithm (left) and the larger space coverage of those returned by our greedy strategy (right). Our greedy strategy guarantees conformational diversity by minimizing the volume overlap of the conformers.

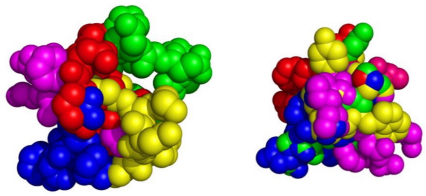


Figure 3 Three fictitious conformers of a protein loop: the surface of the union of the three conformers defines a double torus, i.e. a surface of genus two. Compare to Fig. 2, top row.

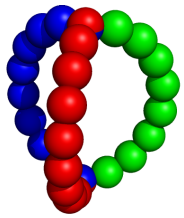
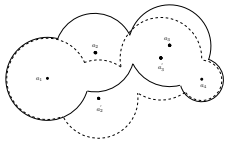


Figure 4 The boundary of the union of balls of the two conformers of Fig. 1, respectively in solid and dashed lines.



2 Selecting Conformers: the Combinatorial Viewpoint

2.1 Arrangements of Balls and Spheres: Volume and Surface Decompositions

The spheres bounding the balls defining 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 *volume arrangement/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 a 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/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 the combinatorial viewpoint –see section 2.3 for the connexion with conformers, assume we are given a base set $\mathcal{U} = \{U_i\}_{i=1,\ldots,m}$ consisting of *cells* (think cells of the volume or surface arrangement), and a collection of *sets* $\mathcal{C} = \{C_i\}_{i=1,\ldots,n}$ (think conformers). 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 set of all subsets of $\mathcal{C}$ of size s.

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

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

For the second class of problems, assume the weight function depends not only on the cells of the decomposition, but also in 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:*

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

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 their 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 an **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 of all possible combinations of conformers is $\mathcal{O}(|\mathcal{C}|^s)$. 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 Volume 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 volume. For conformer selection, optimizing the volume of a selection is a direct way to ascertain a good spatial diversity, since overlaps between conformers are minimized. It is also of interest to drive a simulation process based upon volumetric hydrophobic force fields [Hum99].

Problem 2 from Surface decomposition. Consider the base set $\mathcal{P} = \{P_i\}$ whose cells are those of the $2D$ arrangement. Special cells of this arrangement are those which are exposed, i.e. which define the molecular surface (the VdW surface or the Solvent Accessible Surface depending on the radii used). These exposed patches are of special interest, since they determine the *geometric locus* where interaction occurs between two sets of atoms, in other words the interface [CJ75, CPBJ06]. 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. That is, in Eq. (2), $w_{\mathcal{S}}(P_i)$ stands for some general function defined on the surface patches found on the boundary of the union of balls. For example $w_{\mathcal{S}}(P_i) =$ surface area of patch P_i if P_i is found on the boundary of the union, and 0 otherwise. Notice $w_{\mathcal{S}}(P_i)$ can also be weighted by the radius of its ball, so as to define curvature-based models of hydrophobicity [EM86].

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 method. The greedy algorithm performs s steps, selecting at each step 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 C_j's.

Unfortunately, the selection obtained this way may not realize the optimum solution. As an example consider Fig. 5: for selecting two conformers, the optimum choice has a weight of 14 whereas the greedy method gives us a collection with a weight of 12. To scale this performance, one resorts to the approximation ratio, that is the 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 actually tight [CFN77, NWF78, FG89, Fei98].

2.4.2 Application to Conformer Selections

▷**Volume decomposition, general weight** w**.** Consider a volume 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 section 6, provide generalizations to non-negative weights:

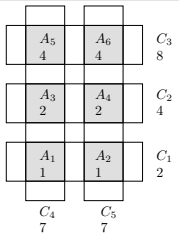
Theorem 2.1. *Consider a volumetric decomposition with non negative weights. For Problem 1, the greedy approach has an approximation guarantee of* $1 - (1 - 1/s)^s > 1 - 1/e$.

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

▷ **Surface decomposition, boundary surface weight** $w_{\mathcal{S}}$**.** For volume decompositions, the previous bound indicates one is always above 63% $(1 - 1/e)$ from the optimum. Unfortunately, as shown in section 7.1, such a result does not hold for problem 2:

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

Figure 5 Selecting two conformers out of $C_1, \ldots, C_5$ by the greedy strategy fails to report the optimal solution. The shaded regions have the weights as indicated and the unshaded regions have null weights.



3 Material and Methods

From now on, as argued in section 1.2 and specified in section 2.3, we focus on the problem of optimizing the MSA of the selection for the particular problem of conformers representing protein loops. The incentives for focusing on the surface area rather than the volume of the conformers are twofold. First, we are not aware of any robust implementation to report the volume of a union of balls. (A mandatory requirement given the density of conformers faces in real test-cases.) A contrario, robust and optimized algorithms exist to handle surface arrangements [CL06, CCLT07, CL07]. Second, as discussed in section 2.3, surface models are ubiquitous in molecular modeling.

3.1 Datasets and Conformer Generation Methods

Protein loops. 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 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]. Complexes 1BTH and 3HHR have been identified as difficult cases since no acceptable structure could be predicted in rigid body docking studies. The Ige Fv Spe7 protein complexed with a recombinant thioredoxin (1OAZ) has been added because of the known flexibility of its interface [JRT03].

The four flexible loops differ by size and degree of variation between the bound and unbound forms, as characterized by the RMSD of the C_α carbons between the bound and unbound forms. In complex 1BTH, the 11 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 12 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 of 1OAZ, the 13 aa loop of the Ige Fv Spe7 protein complexed with a recombinant thioredoxin only undergoes a 2.1 Å deviation, while in 3HHR, the 27 aa loop of the human growth hormone bound to the extracellular domain of its receptor presents a deviations of 5.5 Å.

Conformer generations methods. A number of algorithms exist to generate atomic loop geometries [XSH02, NOS05, LMDL06, SBL07]. We selected Direx [SBL07] and Loopy [XSH02], which respectively generate dense and sparse ensembles of conformers. Each atomic model was subsequently trimmed down to a residue-based model, using the method of [Zac04]. See section 8.2 for the details.

To scale 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.) For atomic and coarse models, the ratio $MSA(Loopy)/MSA(Direx)$ spans the range $[1.79, 4.16]$ and $[1.86, 4.60]$ respectively, which clearly shows the Loopy dataset is much less redundant. See Table 5 in appendix for a full report.

3.2 Greedy Selection: Implementation

The Naive and Priority-based Versions. Denote G_i the selection after i steps, and let R_i stand for the remaining candidates. Following Eq. (2), the naive way of computing G_i consists of incrementally linearly scanning all possible solutions, that is

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

As proved in section 7, the following complexity is worst-case optimal:

Theorem 3.1. *The naive version of Algorithm* `Greedy` *has complexity* $O(ns^3)$.

A more elaborate strategy consists of maintaining 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 in the other conformers. To account for this information, which encodes the complexity of the surface arrangement, denoting $\mathbf{1}_X$ the characteristic function of the Boolean variable X, define

$$\tau = \sum_{C_i \in \mathcal{C}} \sum_{S_j \in C_i} \sum_{P_k} \mathbf{1}_{\text{the ball associated to } S_j \text{contains the patch } P_k || P_k \text{ lies of the surface of } S_j}, \tag{4}$$

where $\mathcal{C}$ is the set of all conformers, S_j a sphere of a conformer C_j and P_k a patch on sphere S_j.

This variant [1], presented in section 7, satisfies:

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

Implementations. The naive implementation was carried out using the `Delaunay_3` and `Alpha_shape_3` packages of the Computational Geometry Algorithms Library [cga]. For the more elaborate one, we used the surface arrangements package described in [CL06, CCLT07, CL07], which is the only one, to the best of out 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 Selection Methods

We compare algorithm `Greedy` against two contenders. The first one, algorithm `HClust`, is a hierarchical agglomerative clustering method based on the average linkage, used for protein-protein docking in [BPZ06]. 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 the coarser (finer) partition features one (n) cluster(s) containing the n (a single) conformers (conformer). As dissimilarity, we use the average RMSD between pairs of conformers across the two clusters. (We also tested the single linkage and complete linkage strategies, which performed equally –data not shown.) 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 is carried out through a two-stage process, namely (i) a fictitious *average* molecule is computed: for k conformers each consisting of p balls centered at $c_{i,j}$, with $i = 1, \ldots, k$ and $j = 1, \ldots, p$, the fictitious molecule consists of p 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 RMSD with this fictitious molecule.

The second one, algorithm `Random`, used as a yardstick, is a mere random shuffling of the conformers, from which a selection of size s is retrieved by taking the first s conformers after shuffling.

3.4 Statistics of Interest: Geometry vs Topology

We first wish to report on the MSA. To see how, for a given selection method M (G: greedy; H: hierarchical; R: random), let $\mathcal{N}_M = \{G_1, \ldots, G_n\}$ be a *collection of selections* of increasing size, i.e. selection G_i contains i conformers. The greedy strategy provides a *nested* collection of selections, since the selection G_{s+1} of size $s+1$ is the selection G_s of size s to which an additional conformer has been prepended. So does the random selection, since we take as selection of size s the first s conformers after the shuffling. The nestedness does

[1] The incentive for presenting two versions is that the winner is not ... that anticipated! See section 4.

not hold for algorithm `HClust`, though. As explained in section 3.3, 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.

To compare two collections of selections, both for the atomic and the coarse models, we report two sets of values. To see which, let R_M be the maximum of the MSA obtained over all selections in $\mathcal{N}_M$, that is $R_M = \max_{G_i \in \mathcal{N}_M} MSA(G_i)$. First, we focus on the maxima of MSA reached, that is on the ratios R_G/R_H and R_G/R_R. Second, denote n_{H_x} (n_{R_x}) the smallest number of conformers required by algorithm H (R) to get a MSA A equal to $x\%$ of its maximum. Also denote n_G the least number of conformers required by the greedy strategy to get a MSA not less than A. We report n_{H_x}/n_G and n_{G_x}/n_G, for $x = 100\%$ and $x = 95\%$.

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. 3 presents and example selection of $g + 1$ conformers anchored at the loops extremities, and defining a genus g surface –$g = 2$ here.

Practically, recall that M_G stands for the maximum MSA obtained with algorithm G, and that $n_{G_{100\%}}$ is the corresponding selection size. We shall compare the variation of β_1 for $n_{G_{100\%}}$ conformers selected by algorithm `Greedy` and `HClust`.

4 Results

Running times. 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. Quite surprisingly, we observed a factor of one order of magnitude in favor of the naive implementation. 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, on all models, the selection of 100 conformers required between 1,000 and 3,000 minutes. Notice though that (i) time is not an issue when the docking algorithm fails (ii) typical time requirements for flexible docking runs vary from a day to one week.

Variations of MSA. A typical variation of the MSA over selections is depicted on Fig. 6, the key figures being summarized in Tables 1 and 2 for the Direx and Loopy datasets respectively. The reader is also referred to section 8.3 for a full report.

Consider first the maximum values reported by the three algorithms. We observe that algorithms `HClust` and `Random` provide comparable performances, a conclusion which holds regardless of the model (atomic or coarse) or the dataset (Direx or Loopy). Algorithm `HClust`, though, tends to peak before algorithm `Random`, as evidenced by the ratios n_H/n_G versus n_R/n_G. When compared to algorithm `Random`, the hierarchical strategy thus essentially provides a gain in convergence rate, at the expense of extra calculations —computing a random permutation is costless. Since our focus in the following is on the MSA rather than the time complexity of the algorithms, we now focus on the comparison of algorithm `HClust` versus algorithm `Greedy`. Speaking of the max values R_G/R_H, one observes an increase in the range 9-13% (Direx, atomic), 10-16% (Direx, coarse), 14-54% (Loopy, atomic) and 29-57% (Loopy, coarse). More interesting is the speed at which the methods peak, as can be seen from the ratios n_{H_x}/n_G and n_{G_x}/n_G, for $x = 100\%$. 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 –decimal values omitted: 10-163 (Direx, atomic), 1-163 (Direx, coarse), 19-79 (Loopy, atomic), 16-67 (Loopy, coarse). On the other hand, as can be seen from the plot Fig. 6, the asymptote is reached rather fast for all algorithms. Focusing on 95% of the max MSA obtained, the ratios $n_{H_{95\%}}/n_G$ now span the following ranges: 1-3 (Direx, atomic), 1-2 (Direx, coarse), 5-28 (Loopy, atomic), 5-20 (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 datasets. This ratio is clearly much higher for Direx than Loopy, which has the following explanation: for a dense dataset 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, even for minute increments available from a redundant dataset. Next, focus on the statistic $n_{H_{95\%}}/n_G$ for the Direx and Loopy datasets. This ratio is much higher for the latter dataset, which shows that algorithm `Greedy` is also better at selecting large increments of MSA within sparse datasets of conformers.

The MSA variation (see plots in section 8.3) also sheds an interesting light on the relative flexibility of the four loops. For the Direx dataset, algorithm `Greedy` peaks in the same way for the four systems. This conclusion does not hold for the Loopy dataset, where the maximum is reached at a slower pace, in particular on model 3HHR. Phrased differently, the 3HHR curve also shows that a collection of $n = 500$ conformers does not provide a full characterization of the conformational space.

Variations of Betti numbers. For a qualitative explanation [2] of these facts, consider the variation of the first Betti number β_1 for the three algorithms. As seen from Tables 3 and

[2]The analysis is qualitative for the following reason: a handle accounts for one unit in the β_1 number, whatever its size. That is a large handle coming from a whole loop (as on Fig. 3) 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. While computing the Betti numbers is by now standard –we use the α-shapes based algorithm of [DE95], the calculation of a geometrically pleasant basis of the homology groups is still an active area of research [CF07].

4, algorithm `Greedy` features a much higher value of β_1 than its contenders. The variation of β_1, illustrated on Fig. 7, is also of interest. All 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, which correspond to a MSA maximization –a property used by algorithm `Greedy`. Once the maximum has been reached, the conformers selected bridge gaps, whence a decrease in β_1. The sharp decreases stops as soon as the union of the selection is essentially a topological ball. The union still features small handles. Such handles get create 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 section 8.3) 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 not enough in the Loopy dataset. 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.

5 Discussion

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 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 paper, we present geometric optimization methods geared towards the characterization and the selection of this 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 provided. Experiments carried out on four protein loops for the optimization of the MSA 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 informa-

tions of the selection (the Betti numbers) yields insights on the quality of the coverage of the conformational space associated to a collection of conformers.

Applications and outlook. Our developments have a number of direct applications. First, our selection methods 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, our selections should also prove useful for flexible docking algorithms resorting to mean field theory. We are currently exploring this issue for difficult protein-protein docking examples, where classical approaches fail from reporting relevant candidate conformations.

Interestingly, this work also raises a number of open 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, there is clear room for improvement.

Table 1 Direx dataset. Comparison of the selection methods. See text for notations.

PDB	$\frac{R_G}{R_H}$	$\frac{R_G}{R_R}$	$\frac{n_{H_{100\%}}}{n_G}$	$\frac{n_{R_{100\%}}}{n_G}$	$\frac{n_{H_{95\%}}}{n_G}$	$\frac{n_{R_{95\%}}}{n_G}$
1BTH-ato.	1.11	1.11	130.0	151.0	3.0	2.0
1CGI-ato.	1.12	1.12	163.33	148.33	2.0	5.33
1OAZ-ato.	1.13	1.13	165.0	165.67	1.33	1.33
3HHR-ato.	1.09	1.09	10.0	6.0	1.0	2.67
1BTH-res.	1.14	1.14	75.8	99.6	1.4	3.4
1CGI-res.	1.12	1.12	163.0	156.0	2.33	2.33
1OAZ-res.	1.16	1.16	1.67	110.33	1.33	3.0
3HHR-res.	1.1	1.09	4.0	3.0	1.67	3.0

Table 2 Loopy dataset. Comparison of the selection methods. See text for notations.

PDB	$\frac{R_G}{R_H}$	$\frac{R_G}{R_R}$	$\frac{n_{H_{100\%}}}{n_G}$	$\frac{n_{R_{100\%}}}{n_G}$	$\frac{n_{H_{95\%}}}{n_G}$	$\frac{n_{R_{95\%}}}{n_G}$
1BTH-ato.	1.21	1.22	33.0	70.57	5.0	21.29
1CGI-ato.	1.14	1.14	79.33	74.5	28.33	52.83
1OAZ-ato.	1.21	1.21	59.13	57.38	20.5	32.0
3HHR-ato.	1.54	1.58	19.7	49.9	14.2	23.7
1BTH-res.	1.29	1.3	44.86	68.57	7.14	34.86
1CGI-res.	1.34	1.35	32.0	82.83	20.5	7.0
1OAZ-res.	1.3	1.3	67.57	71.43	10.14	20.71
3HHR-res.	1.57	1.65	16.0	40.25	5.85	21.08

Table 3 Direx dataset. Comparing the evolution of the first Betti number up to $n_{G_{100\%}}$ selected conformations; 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	9	0.62	1
1CGI-atomic	10	2	12	8.06	8	0	7	0.67	0
1OAZ-atomic	13	5	21	7.07	6	0	19	0.56	0
3HHR-atomic	6	6	30	13.95	11.5	0	23	1.05	0
1BTH-coarse	17	0	12	9.63	11	0	12	0.99	1
1CGI-coarse	22	0	17	6.49	7	0	8	0.93	1
1OAZ-coarse	14	1	18	9.79	10	0	15	1.40	2
3HHR-coarse	11	4	36	16.16	15	1	26	2.10	1

Table 4 Loopy dataset. Comparing the evolution of the first Betti number up to $n_{G_{100\%}}$ selected conformations; 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	2	27	4.91	4
1CGI-atomic	16	4	38	23.68	22	1	21	3	3
1OAZ-atomic	37	7	54	40.09	40	3	26	5.33	3
3HHR-atomic	36	17	342	283.12	306	26	144	80.06	69
1BTH-coarse	32	0	51	36.26	35	1	25	6.85	6
1CGI-coarse	21	0	59	30.15	26	0	31	6.19	6
1OAZ-coarse	28	0	55	40.14	40	4	32	6.66	5
3HHR-coarse	46	1	492	382.59	452	6	202	153.17	150

Figure 6 Loopy dataset. Selections of increasing size for 1BTH atomic: variation of MSA.

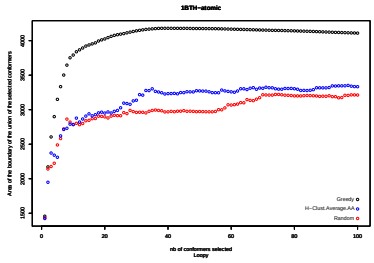
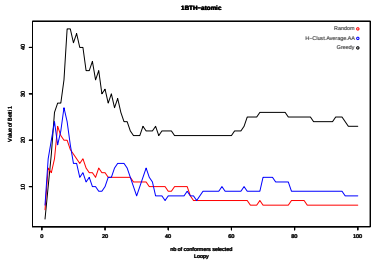


Figure 7 Loopy dataset. Selections of increasing size for 1BTH atomic: variation of the Betti number β_1.



Acknowledgments. The authors wish to thank J. Bernauer, G.F. Schröder, P. Yao on the one hand, and F. Nielsen on the other hand, for insightful discussions about conformer generation methods and optimization, respectively.

References

[AE96] N. Akkiraju and H. Edelsbrunner. Triangulating the surface of a molecule. *Discrete Appl. Math.*, 71:5–22, 1996.

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

[BTLP03] K. Bastard, A. Thureau, R. Lavery, and C. Prevost. Docking macromolecules with flexible segments. *J. of Computational Chemistry*, 24(15):1910–1920, 203.

[CBRL01] H. Claussen, C. Buning, M. Rarey, and T. Lengauer. Flexe: Efficient molecular docking considerating protein structure variations. *JMB*, (308):377–395, 2001.

[CCLT07] P. M. M. De Castro, F. Cazals, S. Loriot, and M. Teillaud. Design of the cgal spherical kernel and application to arrangements of circles on a sphere. Research Report 6298, INRIA, 09 2007.

[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.

[CJ75] C. Chotia and J. Janin. Principles of protein-protein recognition. *Nature*, 256:705–708, 1975.

[CL06] F. Cazals and S. Loriot. Computing the exact arrangement of circles on a sphere, with applications in structural biology. Research Report 6049, INRIA, 12 2006. `https://hal.inria.fr/inria-00118781`.

[CL07] F. Cazals and S. Loriot. Computing the exact arrangement of circles on a sphere, with applications in structural biology. In *Proc. 23th Annu. ACM Sympos. Comput. Geom. —Video/Multimedia track*, 2007.

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

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

[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.

[CSdA+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. In *ISMB*, 2005.

[DE95] C.J.A. Delfinado and H. Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Comput. Aided Geom. 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 *WABI*, pages 265–276, 2007.

[EM86] D. Eisenberg and A.D. McLachlan. Solvation energy in protein folding and binding. *Nature*, 319:199–203, 1986.

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

[FG89] P.C. Fishburn and W.V. Gehrlein. Pick-and choose heuristics for partial set covering. *Discrete Appl. Math.*, 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.

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

[Hum99] G. Hummer. Hydrophobic force field as a molecular alternative to surface-area models. *J. Am. Chem. Soc.*, 121:6299–6305, 1999.

[JRT03] L.C. James, P. Roversi, and D.S. Tawfik. Antibody multispecificity mediated by conformational diversity. *Science*, 2999:1362 1367, 2003.

[JW07] J. Janin and S. Wodak. The third capri assessment meeting, toronto, canada, april 20-21, 2007. *Structure*, 2007.

[KD94] P. Koehl and M. Delarue. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J. Mol. Biol.*, 239, 1994.

[KD95] P. Koehl and M. Delarue. A self consistent mean field approach to simultaneous gap closure and side-chain positioning in homology modeling. *Nature Struct. Biol.*, 2:163–170, 1995.

[LMDL06] G. Liu, J. Milgram, A. Dhanik, and J.C. Latombe. On the inverse kinematics of a fragment of protein backbone. In *10th Symp. on Advances in Robot Kinematics*, Ljubljana, Slovenia, 2006.

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

[NOS05] K. Nooman, D. O'Brien, and J. Snoeyink. Probik: protein backbone motion by inverse kinematics. *Int. J. Robotics Research*, 24(11), 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.

[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, December 2007.

[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. *PNAS*, page 102179699, 2002.

[Zac04] M. Zacharias. Rapid protein-ligand docking including soft degrees of freedom from molecular dynamics simulations to account for protein flexibility: Fk506 binding to fkbp binding protein as an example. *Proteins*, 54:759–767, 2004.

6 Supplement: Volume Decompositions

6.1 Greedy: Approximation Factor and Optimality

6.1.1 Approximation Factor

We shall use the following notations. The conformer selected at the kth 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 $C_j, 1 \leq j < k$. We need the following lemma in order to prove the theorem.

Lemma 6.1. *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 elements A_i being covered. The weight of the elements 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 elements involved in Eq. (6) are covered by the optimum set of conformers, by the union-bound property, there must exist a conformer that covers new elements 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 elements 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 6.1. As a counter-example, consider the sets $C_1 = \{e1, e2\}, s2 = \{e2, e3\}$ with $w(e1) = w(e3) = 1$ and $w(e2) = -1$. The union-bound fails for $w(C_1 \cup C_2)$.

Using Lemma 6.1, the proof of Thm. 2.1 goes as follows:

Proof. Thm. 2.1 Multiplying the inequality obtained for step one 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. □

6.1.2 Optimality

To prove Thm. 2.1, we construct tight examples for the greedy approach.

Proof. Thm. 2.1 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

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

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

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

The conformers are defined as follows

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

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

$$\forall 1 \leq i \leq s,\ C_{s+i} = \bigcup_{j \equiv i \pmod{s}} A_j$$

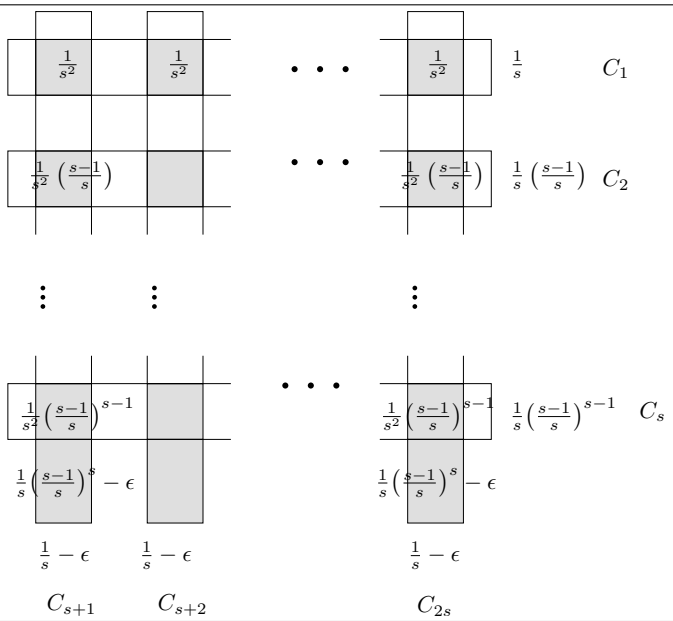
Simple calculations lead us the following total weights for the conformers

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

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

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 arbitrarily close to $1 - (1 - \frac{1}{s})^s$. □

Figure 8 A tight example for the greedy strategy



7 Supplement: Surface Decompositions

7.1 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 $1/s^2$, □

7.2 Naive Algorithm for Surface Arrangement

Proof of Thm. 3.1:

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 balls, this is done in worst-case optimal time of $O(i^2)$, 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 $\sum_{i=1}^{s}(n - i + 1)O(i^2)$, whence the claim. □

7.3 Priority-based Algorithm for Surface Arrangement

Notations. 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 notations, 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 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 [CL06, CCLT07, CL07], 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. 9.

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 hashset 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 $O(\tau + s \log n)$. □

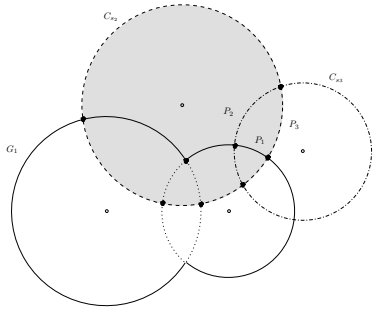
Algorithm 1 Greedy algorithm for surface decomposition.

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



8 Supplement: Material and Methods

8.1 Conformers Generation methods: Direx versus Loopy

8.2 Direx and Loopy

Upon generation of an atomic protein model involving a given loop conformer, a coarse protein conformer is obtained applying the reduction of [Zac04]. In this coarse model, each ball represents three or four heavy atoms of the protein, and there are between one and three beads per amino acid. One of these beads represents the main chain atoms and is centered at the C_α, one bead is centered at the C_β atom and, when appropriate, a third bead is placed at the geometric center of the remaining heavy atoms of the side chain. The radius of each bead used in the coarse grain model has been defined based on the van der Waals radius of the atoms entering in its composition.

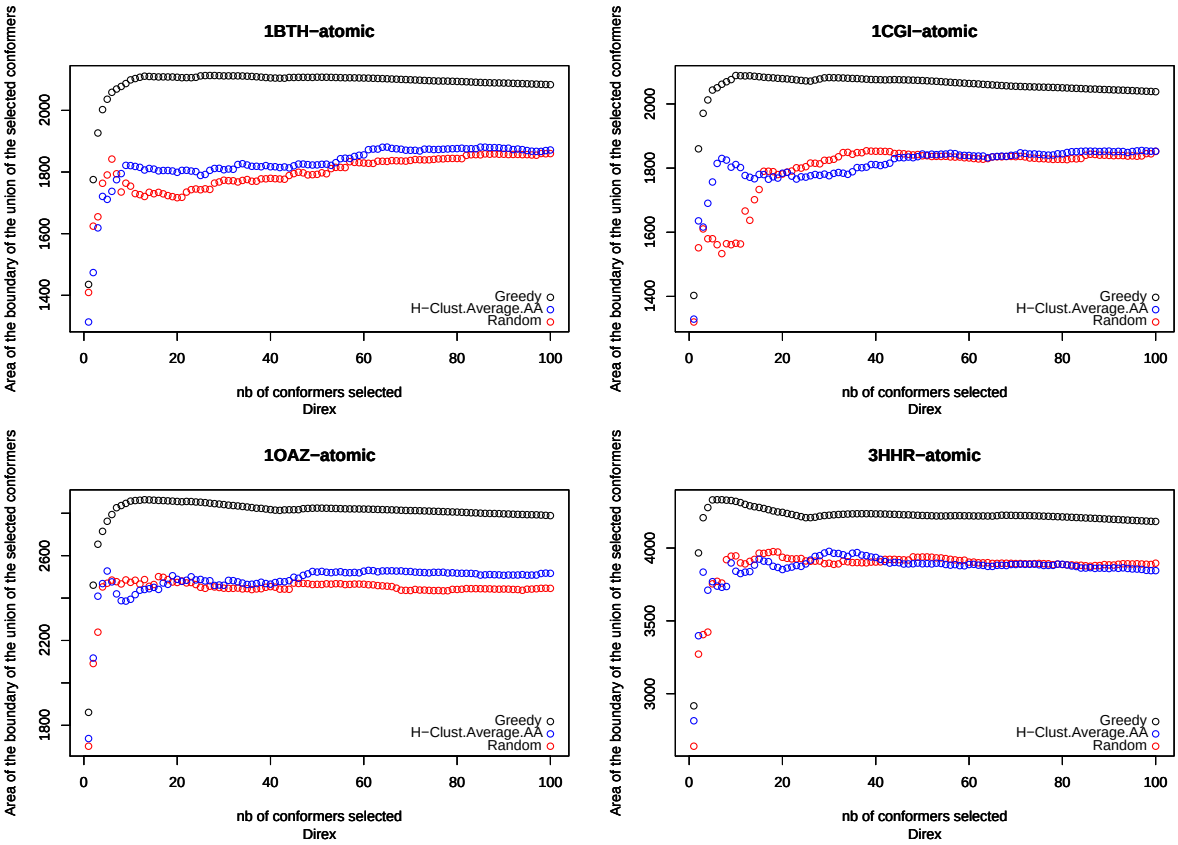
We selected two algorithms to generate loop conformers, which respectively yield dense and sparse ensembles of conformers. Algorithm Direx [SBL07], based on algorithm CONCOORD [dGvAS$^+$97], handles a whole protein and processes all the atoms in the same way. The method consists of performing perturbations of the atomic positions while preserving constraints on internal coordinates (bond lengths and dihedral angles). The generation of n conformers is greedy, 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.

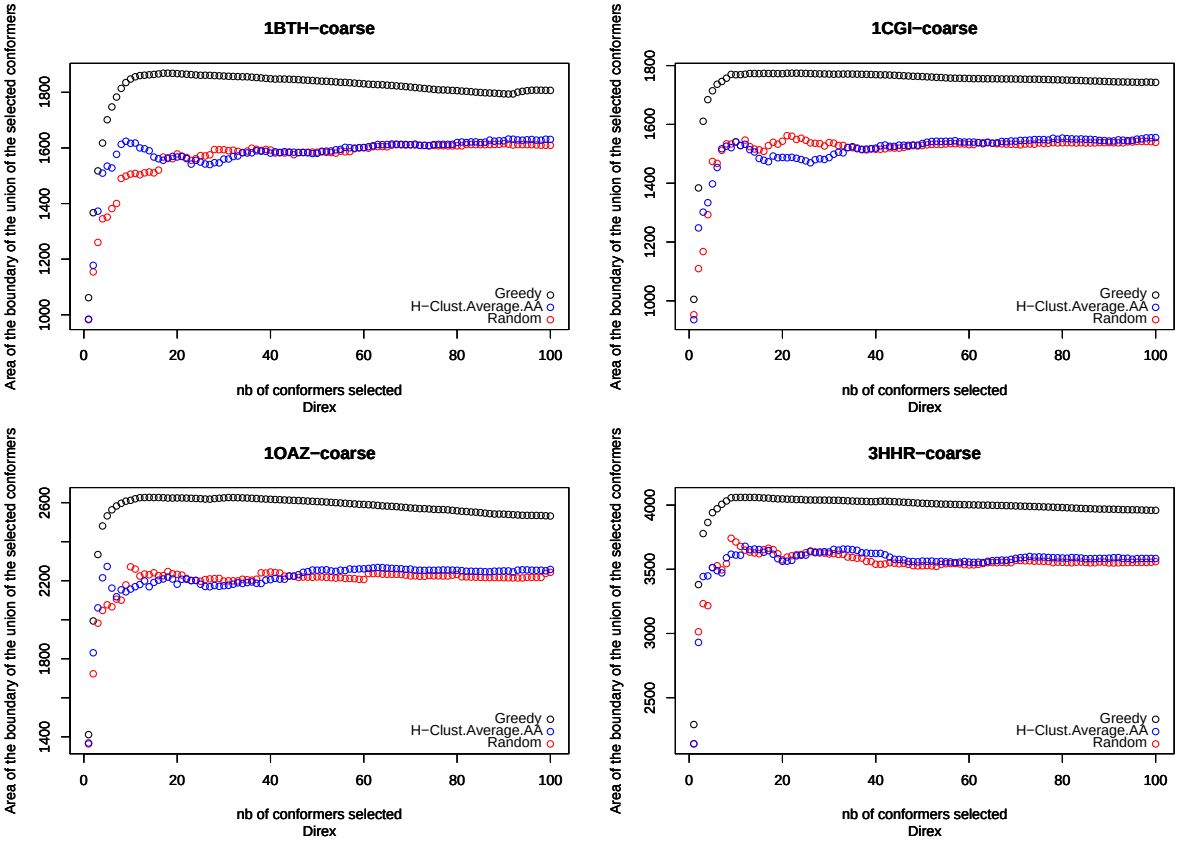
Algorithm Loopy [XSH02] is a genetic-like algorithm which consists of evolving a population of loops, the $k+1$th generation mixing a subset (survivors) of the kth generation together with new individuals derived from this subset. The main features of the algorithm are two-fold. First, the algorithm 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.

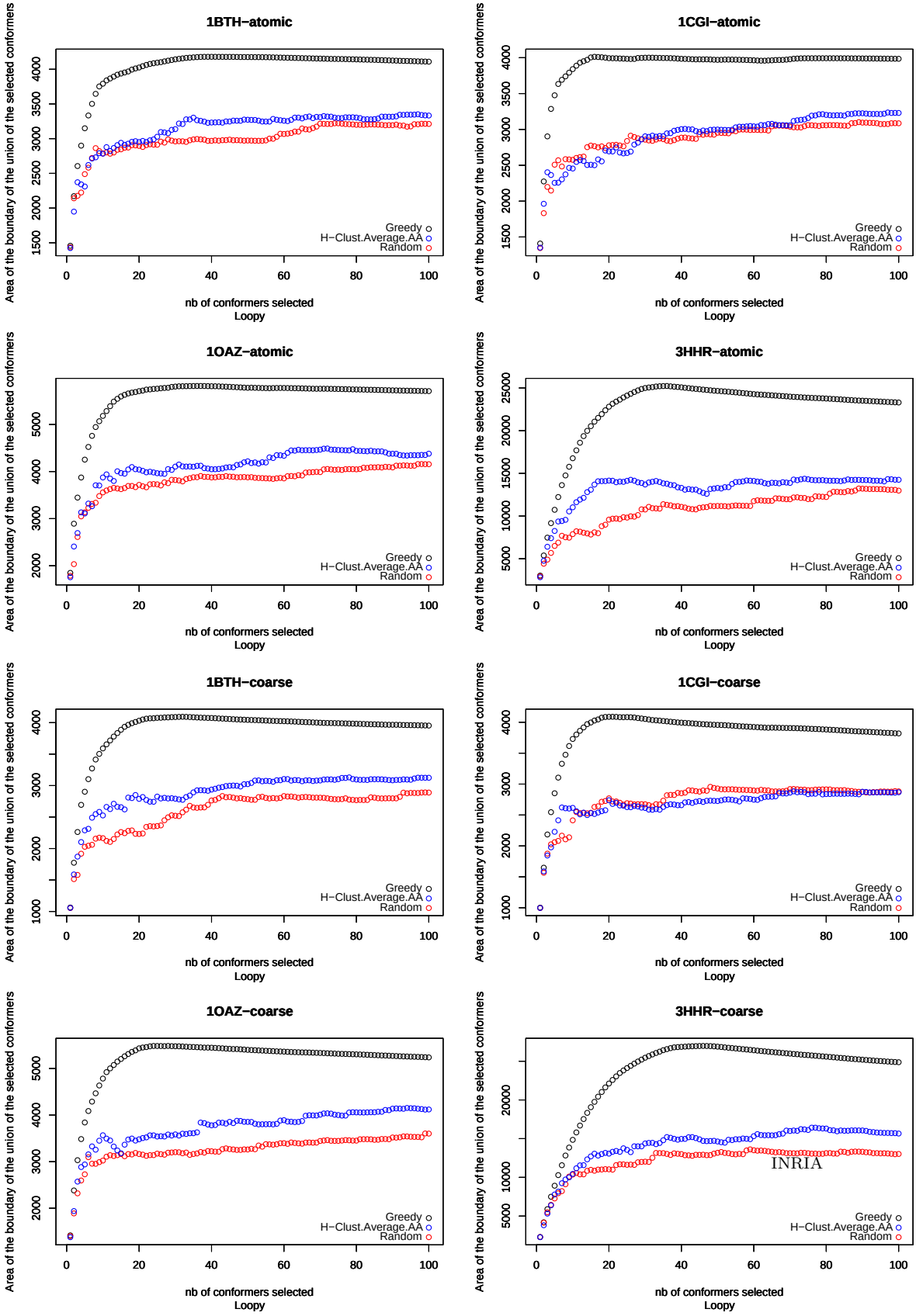
Table 5 Direx versus Loopy: MSA and volume for $n = 500$ conformers. The number of residues comprises the two residues bounding all the conformers –these are common to all conformers.

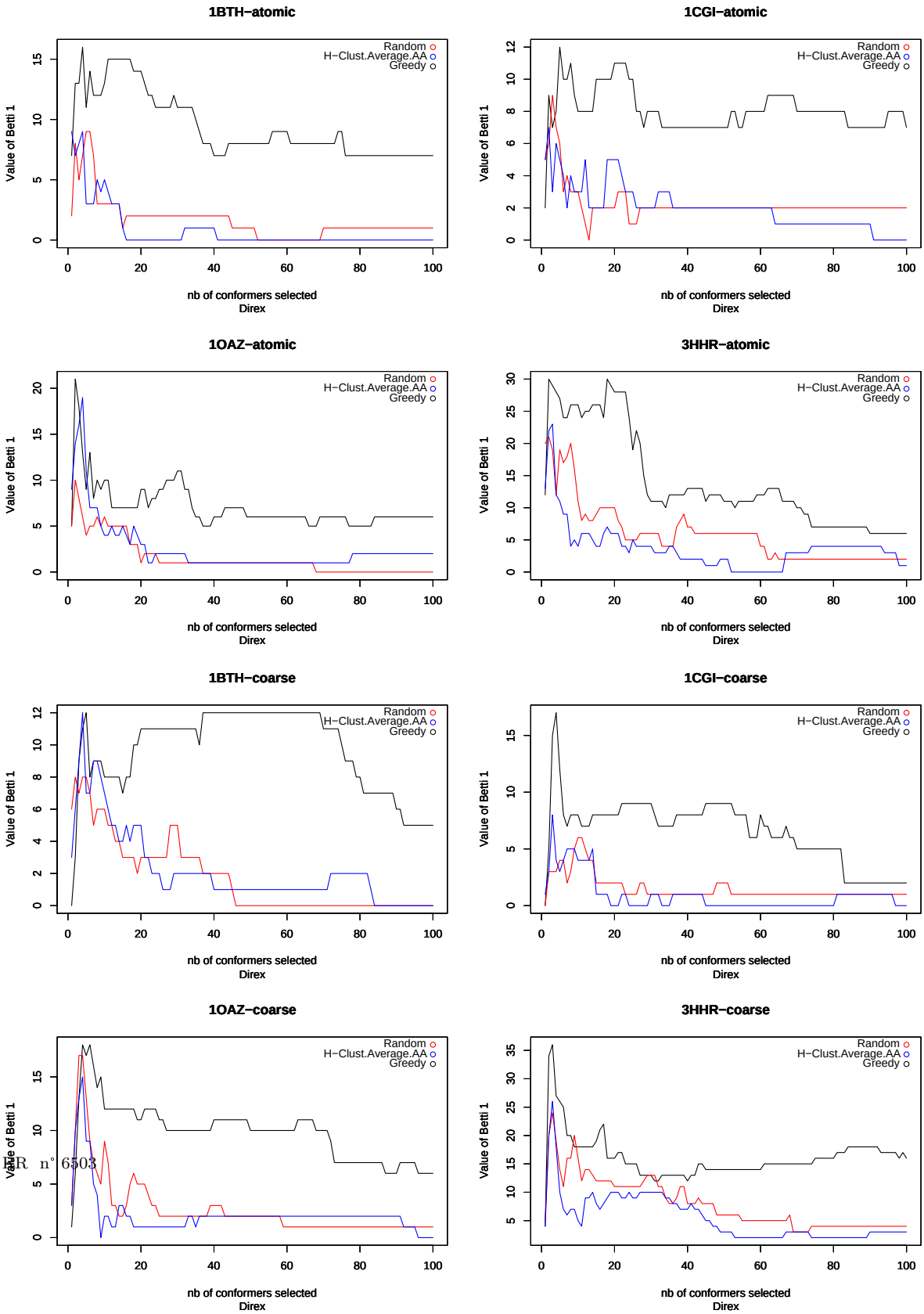
PDBCODE	MSA Direx	MSA Loopy	Volume Direx	Volume Loopy	Nb balls	Nb res.	RMSD
1BTH-atomic	1906.01	3423.34	5041.13	9740.27	108	11	5.7Å
1BTH-coarse	1639.72	3142.11	3592.87	8119.68	29	11	
1CGI-atomic	1867.17	3516.97	4737.87	9853.29	103	12	unres.
1CGI-coarse	1583.7	3032.78	3324.03	8169.20	28	12	
1OAZ-atomic	2535.6	4788.63	6843.13	14323.00	142	13	2.1Å
1OAZ-coarse	2262.51	4211.13	5568.03	12143.20	37	13	
3HHR-atomic	3835.2	15976.8	10448.5	54618.50	223	27	5.5Å
3HHR-coarse	3549.63	16345.7	8520.82	48339.70	67	27	

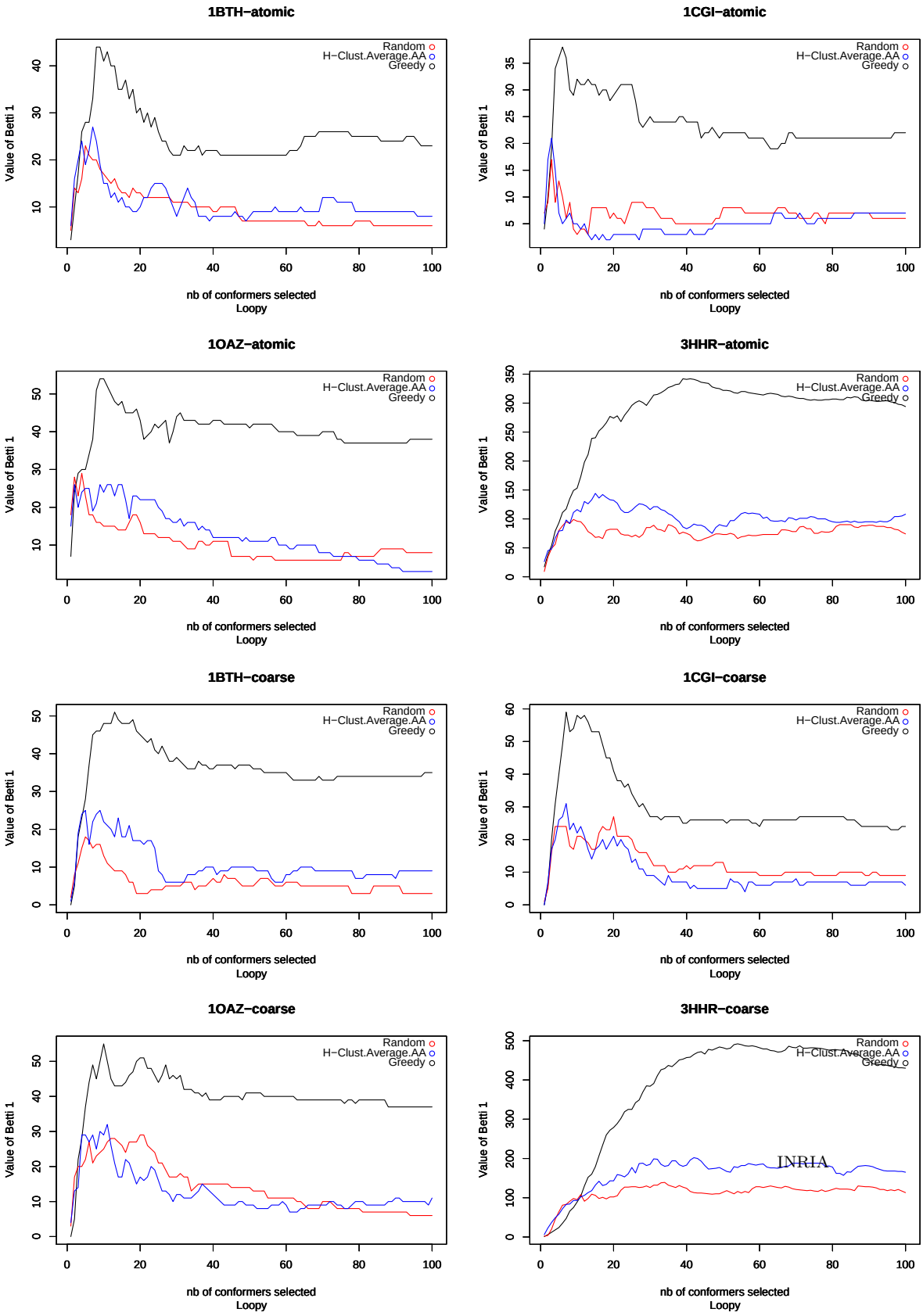
8.3 Graphs


1BTH−atomic
1CGI−atomic
1OAZ−atomic
3HHR−atomic
Area of the boundary of the union of the selected conformers
nb of conformers selected
Direx
Greedy
H−Clust.Average.AA
Random


1BTH−coarse
1CGI−coarse
1OAZ−coarse
3HHR−coarse
Area of the boundary of the union of the selected conformers
nb of conformers selected
Direx
Greedy
H−Clust.Average.AA
Random


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
H−Clust.Average.AA
Random
INRIA


1BTH−atomic
1CGI−atomic
1OAZ−atomic
3HHR−atomic
1BTH−coarse
1CGI−coarse
1OAZ−coarse
3HHR−coarse
Random
H−Clust.Average.AA
Greedy
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
Random
H−Clust.Average.AA
Greedy
Value of Betti 1
nb of conformers selected
Loopy

Contents

1 Introduction **4**
1.1 Flexibility in Docking Methods 4
1.2 Contributions and Paper Overview 5

2 Selecting Conformers: the Combinatorial Viewpoint **7**
2.1 Arrangements of Balls and Spheres: Volume and Surface Decompositions . . 7
2.2 Optimization Problems 7
2.3 Instantiations to Conformer Selection 8
2.4 The Greedy Strategy 9
2.4.1 The strategy and its Guarantees 9
2.4.2 Application to Conformer Selections 9

3 Material and Methods **10**
3.1 Datasets and Conformer Generation Methods 10
3.2 Greedy Selection: Implementation 11
3.3 Selection Methods 12
3.4 Statistics of Interest: Geometry vs Topology 12

4 Results **13**

5 Discussion **15**

6 Supplement: Volume Decompositions **21**
6.1 Greedy: Approximation Factor and Optimality 21
6.1.1 Approximation Factor 21
6.1.2 Optimality 22

7 Supplement: Surface Decompositions **24**
7.1 Approximating Factor for Problem 2 24
7.2 Naive Algorithm for Surface Arrangement 24
7.3 Priority-based Algorithm for Surface Arrangement 24

8 Supplement: Material and Methods **28**
8.1 Conformers Generation methods: Direx versus Loopy 28
8.2 Direx and Loopy 28
8.3 Graphs 30

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