Protein Packing Quality Using Delaunay Complexes

Rasmus Fonseca
Dept. of Computer Science
University of Copenhagen
Copenhagen, Denmark
Email: rfonseca@diku.dk

Pawel Winter
Dept. of Computer Science
University of Copenhagen
Copenhagen, Denmark
Email: pawel@diku.dk

Kevin Karplus
Biomolecular Engineering Dept.
University of California
Santa Cruz, USA
Email: karplus@soe.ucsc.edu

***Abstract*—A new method for estimating the packing quality of protein structures is presented. Atoms in high quality protein crystal structures are very uniformly distributed which is difficult to reproduce using structure prediction methods. Packing quality measures can therefore be used to assess structures of low quality and even to refine them.**

Previous methods mainly use the Voronoi cells of atoms to assess packing quality. The presented method uses only the lengths of edges in the Delaunay complex which is faster to compute since volumes of Voronoi cells are not evaluated explicitly. This is a novel application of the Delaunay complex that can improve the speed of packing quality computations. Doing so is an important step for, e.g., integrating packing measures into structure refinement methods. High- and low-resolution X-ray crystal structures were chosen to represent well- and poorly-packed structures respectively. Our results show that the developed method is correlated to the well-established RosettaHoles2 but three times faster.

***Keywords*-Delaunay complex; protein; packing quality;**

I. Introduction

The resolution of a protein structure indicates how accurate the experimentally determined positions of atoms are in the protein. Protein structures with resolutions less than 1.8Å are generally considered good and they are, paradoxically, referred to as high-resolution structures. High-resolution structures are characterized by a uniform distribution of atoms in the core. Low-resolution structures and structures solved partially or wholly by computational methods tend to form clusters of atoms in some places and holes or voids in others. This is referred to as bad packing of the atoms. An estimate of the packing quality can be used to improve structure assessment software such as WHAT-CHECK [1], PROCHECK [2] or ProSA [3]. Also, it can be added as an additional term in free energy functions used in protein structure prediction or refinement.

A number of methods have been developed to characterize packing [4], [5], many of which use the volumes of Voronoi cells for atoms [6], [7], [8]. A very recent and popular method is RosettaHoles2 [9], [10]. This method uses the Voronoi diagram and a support vector machine to output a packing energy. For each atom, twenty spheres with increasing radii are centered on the atom. The volumes of the intersections between the spheres and the Voronoi cell of the atom are used as input features to the support vector machine. The number characterizing the packing, the *RosettaHoles2 cost*, is found by averaging the output of the support vector machine for all atoms.

We use the Delaunay complex of all heavy (non-hydrogen) atoms to quantify the packing quality of protein structures. The Delaunay complex, $DC(A)$, of a set of points, A, consists of all 3-simplices (tetrahedra) whose circumsphere does not contain a point of A in its interior, as well as all faces of simplices in $DC(A)$. The 1-simplices in $DC(A)$ are a set of edges between points of A. In this study we assume that all atoms have roughly the same radii and hence can be represented by a set of points. The packing quality is found using only the edges of $DC(A)$ as input features to a feed-forward neural network. Because the faces of the Voronoi cell intersect the edges of the Delaunay complex at their midpoints, the lengths of edges roughly capture the geometry of the Voronoi cell. However, much less computation is required when the cell volume is not explicitly calculated. For training purposes, high-resolution and low-resolution structures are used to represent well-packed and poorly-packed structures respectively.

Our method distinguishes itself from other methods in two ways. First, it uses the edges of the Delaunay complex and therefore does not require volume calculations of the Voronoi cells. Second, only low-resolution X-ray structures are chosen to represent poorly-packed molecules. This

is in contrast to RosettaHoles2, where predicted structures generated by Rosetta [11] are also included. There exist many scoring methods that separates predicted structures from native structures, but poor packing is one of the things that often distinguishes low-resolution structures from high-resolution ones. The main conclusion of this paper is that the edges of the Delaunay complex characterize packing as well as the volume integration of the Voronoi cell used in RosettaHoles2, but can be computed faster.

II. Methods

The output of the method described here is a *packing cost* which is a quantification of the packing quality of a protein structure. The packing cost of a structure is the average *atom packing cost* of the individual atoms in the structure. The following section describes how the atom packing cost is calculated and why averaging atom packing costs to get the packing cost is reasonable. Finally, the data sets used for training and testing are described.

First, the Delaunay complex of the centers of all heavy atoms, A, is found using the insertion algorithm described by Ledoux [12]. Although this algorithm has a $\mathcal{O}(n^2)$ worst-case running time (where $n = |A|$), in practice it runs fast for two reasons. First, the atoms are inserted in the order they appear in the protein chain. When searching for the tetrahedron containing the inserted point, the method walks from an adjacent tetrahedron of the previously inserted point and, in practice, only traverses a constant number of tetrahedra. Second, the method uses flipping to reinstate the Delaunay criterion after a point is inserted. Since the flipping only affects tetrahedra whose circumsphere contains the newly inserted point, insertion is, in practice, a constant-time operation for evenly distributed points. Assuming that both the point-location and reinstating the Delaunay criterion are expected $\mathcal{O}(1)$ time operations, the algorithm runs in expected $\mathcal{O}(n)$ time.

We define an atom to be *buried* if none of its adjacent tetrahedra are exposed. A tetrahedron, τ, is *exposed* iff there exist a sequence of adjacent tetrahedra, all with circumradii larger than 2.4Å, starting at τ and ending at a tetrahedron which has a face on the convex hull. The radius of 2.4Å is often used as the combined radii of an average heavy atom and a water molecule. Therefore, if a tetrahedron is exposed it indicates that a water molecule can gain access to its interior.

An atom packing cost is assigned to each heavy atom using a feed-forward neural network with 10 input neurons, 20 hidden neurons and 1 output neuron. The values assigned to the input neurons are based on the lengths of edges incident to the atom in the Delaunay complex. Ten bins are defined as shown in Table I. The value of an input neuron is the number of incident edges whose length fall within that bin. When training, the desired atom packing cost for the neural network is 0 if the atom is in a structure with resolution less than 1.8Å and 1 otherwise. The actual output of the neural network is the atom packing cost. Because different types of atoms (carbon, nitrogen, oxygen and sulfur) might appear in different contexts within a protein, a separate neural network is trained for each of the four types of atoms. Sulfur, for instance, has a significantly larger radius than either of the other three atom types. Edges adjacent to a sulfur atom will therefore typically be longer, which is not necessarily an indication of bad packing.

Bin	Interval
0	$[0, 1.15)$
1	$[1.15, 2.04)$
2	$[2.04, 2.13)$
3	$[2.13, 2.44)$
4	$[2.44, 2.72)$
5	$[2.72, 3.01)$
6	$[3.01, 3.34)$
7	$[3.34, 3.71)$
8	$[3.71, 4.13)$
9	$[4.13, \infty)$

Table I
THE INTERVALS OF BINS USED FOR THE 10 INPUTS IN THE NEURAL NETWORK.

The intervals of the bins in Table I are calculated such that any edge incident to a buried atom in the training set has an equal probability of being in any of the bins. To determine these intervals, all edges incident to buried atoms in the training set (defined briefly) are collected in a list. This list is sorted according to the lengths of the edges, and split in ten lists of equal sizes. The last elements of the 9 first lists are used as the boundaries of the bin-intervals.

When training the neural networks to output the atom packing cost, only buried atoms are used as training examples. The reason is that the network might be trained to recognize the size of the protein instead of the packing quality. Non-buried atoms have long adjacent edges, and can be recognized by the number of edges in bin 9. If a neural network is accidentally trained to recognize

the number of non-buried atoms it will have an estimate of the surface area and hence the size of the protein. This is a problem because the average size of low-resolution structures is larger than for high-resolution.

When evaluating the packing cost of a structure, the average atom packing costs of *all* atoms is returned. Averaging over buried atoms only does not significantly affect the packing cost, and since it takes longer to determine which atoms are buried than to calculate the atom packing cost of all atoms, the latter is chosen. Averaging atom costs is justified by inspecting the distribution of atom packing costs. For most proteins this distribution roughly follows a normal distribution which is defined by an average and a standard deviation.

A *training set*, consisting of 3982 protein structures, is retrieved from the PISCES server [13] (pre-compiled data set id: cullpdb_pc40_res3.0_R1.0_d110218). No two structures within this set have sequence similarity higher than 40%. Half of the structures in the training set, the high-resolution structures, have a resolution less than 1.61Å. The other half, the low-resolution structures, have a resolution greater than 2.24Å. All chains that are not specified by the PISCES server are disregarded even though they appear in PDB-files necessary for the test and training sets. Ligands and other heterogeneous atoms (HETATM records) are included and atoms with multiple occupancies are filtered such that only the atom with highest occupancy is included. Only chains with 50 amino acids or more are included. The training set is the basis for all the choices made in the packing cost method and it is used to train the four neural networks.

A *test set*, consisting of 1838 protein structures, is retrieved from the PISCES server such that no two structures in the training set and the test set have more than 40% sequence similarity. As in the training set, half of the structures are high-resolution and the other half are low-resolution. The PDB-files are treated in the same way those in the training set. The test set is used to determine if the packing cost can successfully discriminate between high- and low-resolution structures and is also the basis for the timing experiments in the Results section.

The *CASP9 set*, consisting of all 49899 protein structures submitted to the CASP9 experiment, is retrieved from predictioncenter.org. These structures are examples of computationally generated structures similar to those used as examples of bad packing in RosettaHoles2. This data set is used to confirm the hypothesis that computational structures are poorly-packed and to compare the packing cost to the RosettaHoles2 cost.

III. Results

The experiments seek to illustrate that the packing cost discriminates between well-packed and poorly-packed structures as well as RosettaHoles2, but does it faster.

The discriminatory power of the packing cost is illustrated using distributions of packing costs. Figure 1 shows distributions of packing costs for high- and low-resolution structures in the test set and for structures in the CASP9 set. Figure 2 shows similar distributions for the RosettaHoles2 cost.

The neural networks that determine the packing cost are trained to distinguish high-resolution structures from low-resolution structures so it may seem surprising that the corresponding distributions in Figure 1 are not completely separated. The differences between high- and low-resolution structures can be very subtle so sometimes the packing cost will mis-categorize. As expected, however, most high-resolution structures have a lower packing cost than low-resolution structures and the degree of misclassification is not worse than that of the RosettaHoles2 cost, shown in Figure 2.

Both the packing cost and RosettaHoles2 cost can separate high-resolution structures from CASP9 structures with a high accuracy. This is noteworthy because, unlike the RosettaHoles2 cost, the packing cost is not trained specifically to classify computationally generated structures.

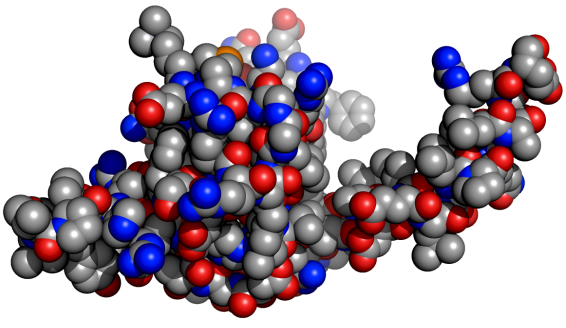

Figure 4. Typical example of a structure with very high RosettaHoles2 cost.

The packing cost and RosettaHoles2 cost both separate high-resolution structures from computer-generated ones, but they may characterize different

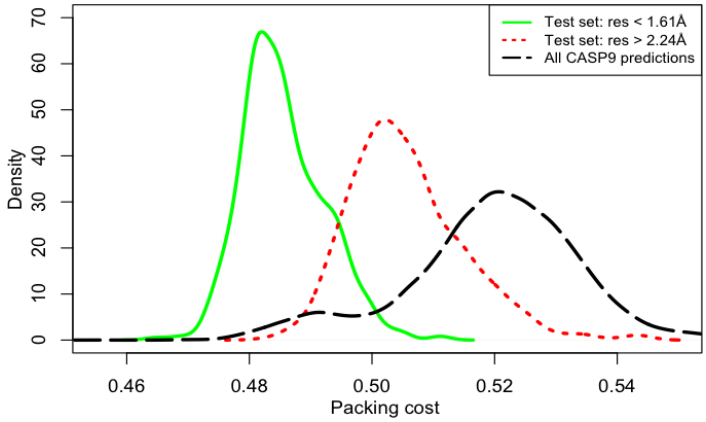

Figure 1. Distributions of packing costs for proteins in the test set and the CASP9 set.

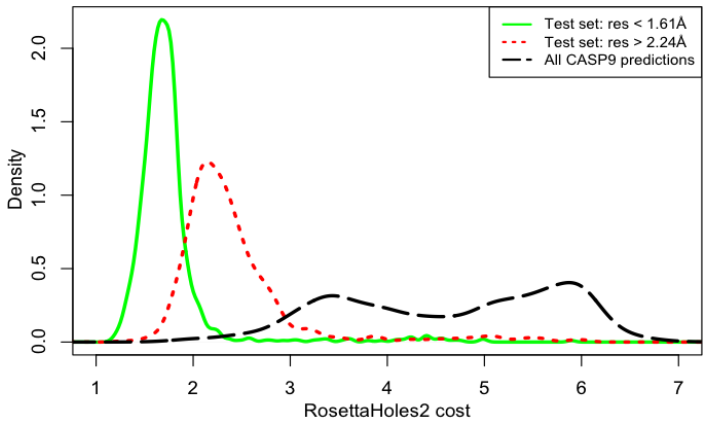

Figure 2. Distributions of RosettaHoles2 costs for proteins in the test set and the CASP9 set.

properties. Figure 3 shows a scatter-plot of RosettaHoles2 costs plotted against packing costs. The main cluster of structures has RosettaHoles2 costs between 1 and 3. Within this cluster there is a clear linear correspondence between the packing cost and the RosettaHoles2 cost (Pearson's squared r of 0.65). There are roughly 100 structures with a RosettaHoles2 cost of more than 3.0. The majority of these are non-globular chains, often with an extended and exposed piece as shown in Figure 4. It is not clear if such structures should be considered well-packed since they are not complete, so it is chosen to disregard these. There are also 15 structures with RosettaHoles2 costs less than 1. It seems that ligands or residues marked as 'unknown' are responsible for most of these, since removing them causes the RosettaHoles2 cost to increase above 1. These are disregarded as well. It is noted that the packing cost is very robust and never returns very extreme values. It is also observed that for the majority of proteins, there is a correlation between the packing cost and the RosettaHoles2 cost.

To demonstrate the improved speed of our method, the system time of the packing cost calculation is measured and displayed as a function of the number of atoms in each structure (Figure 5). The same is done for RosettaHoles2. Both programs are run on a MacBook 2GHz computer and the timing is performed in the source code with `getrusage`. Only the system time of the scoring itself, and not, for example, the time to read the PDB-file, is measured.

For the smallest proteins with less than 500 atoms, the packing cost is calculated between 3 and 4 times faster than the RosettaHoles2 cost. For the larger proteins with roughly 6000 atoms, the packing cost is calculated more than 5 times faster. The computation that dominates out method is finding the Delaunay complex. As mentioned in the Methods section the insertion algorithm uses the chain-structure of the protein to generate

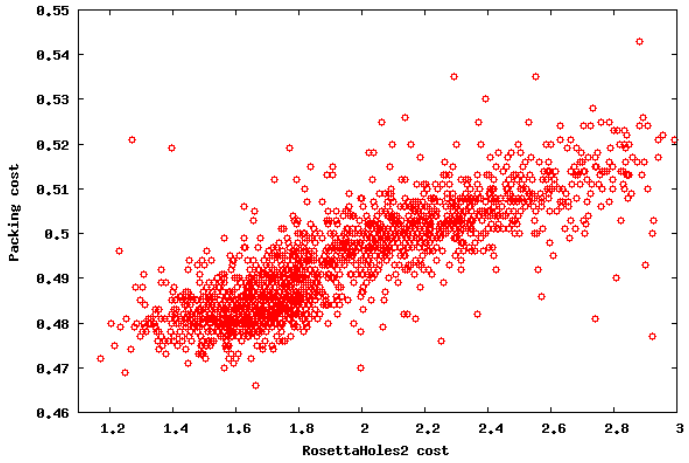

Figure 3. Correlation between packing cost and RosettaHoles2 cost for proteins in the test set.

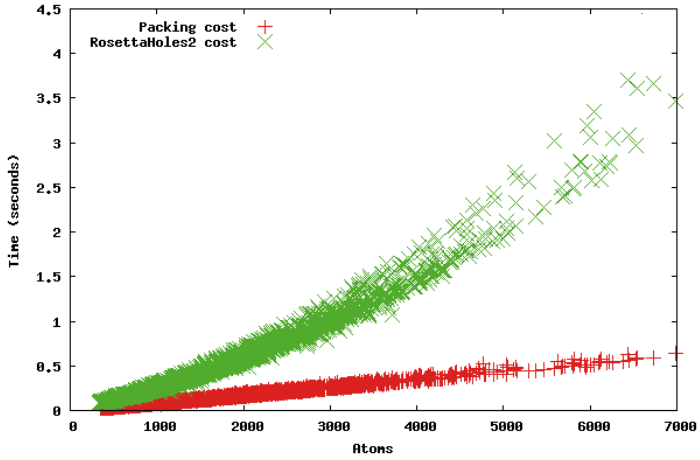

Figure 5. Timing of the packing cost and the RosettaHoles2 cost for proteins in the test set.

the Delaunay complex in expected linear time. This fact is clearly reflected in the timing plot on Figure 5. RosettaHoles2 uses the DAlphaBall program [14], [15] to get the volumes of Voronoi cells. As an intermediate step DAlphaBall finds the Delaunay complex using an insertion and flipping algorithm similar to ours, but it contains a data structure for point-location which gives an expected running time of $\mathcal{O}(n \lg n)$ and does not utilize the chain-structure of proteins.

The ultimate goal of having a fast characterization of the packing cost is to include it as a term in an energy function and improve the packing quality of a protein structure computationally. For a typical protein of $\approx$ 2000 atoms, the packing cost is calculated in $\approx$ 200ms which, in theory, is fast enough to do structure refinement on a massively parallelized system. Furthermore there are a number of ways to improve the speed of the packing cost. Lui and Snoeyink [16], e.g., reports a running time of the tess3 triangulation program that is at least 3 times faster than our insertion algorithm. Guibas and Russel [17] describes how updating the Delaunay complex, after a subset of the points have moved, can be performed faster than recalculating the entire Delaunay complex.

A problem with the packing cost is that many energy functions (Rosetta's, for instance) require their energy terms to be differentiable in order to do fast updates of the energy. In its current form the packing cost is not differentiable. One of the main findings of this paper, however, is that edge-lengths in the Delaunay complex characterize packing just as well as the volume of the Voronoi cells. Since the edge-lengths can easily be differentiated with respect to vertex-

coordinates one can create a differentiable packing cost measure by using a differentiable machine learning method such as support vector machines on distributions of edge-lengths.

IV. Conclusion

An estimate of the packing quality is useful for computational refinement of protein structures. A packing cost was developed and shown to characterize the packing quality of proteins. It was concluded that using edges of the Delaunay complex for characterizing packing is just as efficient as using the Voronoi cells. The observed improvements in speed over previous methods makes it well suited for integration into an energy function.

Acknowledgements

We thank William Sheffler for his kind help in making RosettaHoles2 run properly.

References

[1] R. W. W. Hooft, G. Vriend, C. Sander, and E. E. Abola, "Errors in protein structures," *Nature*, vol. 381, no. 6580, p. 272, 1996.

[2] R. A. Laskowski, M. W. MacArthur, D. S. Moss, and J. M. Thornton, "PROCHECK: a program to check the stereochemical quality of protein structures," *Journal of Applied Crystallography*, vol. 26, no. 2, pp. 283–291, 1993.

[3] M. J. Sippl, "Recognition of errors in three-dimensional structures of proteins," *Proteins*, vol. 17, no. 4, pp. 355–362, 1993.

[4] N. Pattabiraman, K. B. Ward, and P. J. Fleming, "Occluded molecular surface: analysis of protein packing." *Journal of Molecular Recognition*, vol. 8, no. 6, pp. 334–344, 1995.

[5] J. M. Word, S. C. Lovell, T. H. LaBean, H. C. Taylor, M. E. Zalis, B. K. Presley, J. S. Richardson, and D. C. Richardson, "Visualizing and quantifying molecular goodness-of-fit: small-probe contact dots with explicit hydrogen atoms." *Journal of Molecular Biology*, vol. 285, no. 4, pp. 1711–1733, 1999.

[6] M. Gerstein, J. Tsai, and M. Levitt, "The volume of atoms on the protein surface: calculated from simulation, using Voronoi polyhedra." *Journal of Molecular Biology*, vol. 249, no. 5, pp. 955–966, 1995.

[7] A. Poupon, "Voronoi and Voronoi-related tessellations in studies of protein structure and interaction." *Current Opinion in Structural Biology*, vol. 14, no. 2, pp. 233–241, 2004.

[8] K. Rother, P. W. Hildebrand, A. Goede, B. Gruening, and R. Preissner, "Voronoia: analyzing packing in protein structures," *Nucleic Acids Research*, vol. 37, no. suppl 1, pp. D393–D395, 2009.

[9] W. Sheffler and D. Baker, "RosettaHoles: Rapid assessment of protein core packing for structure prediction, refinement, design, and validation," *Protein Science*, vol. 18, no. 1, pp. 229–239, 2008.

[10] ——, "Rosettaholes2: A volumetric packing measure for protein structure refinement and validation," *Protein Science*, vol. 19, no. 10, pp. 1991–1995, 2010.

[11] C. A. Rohl, C. E. M. Strauss, K. Misura, and D. Baker, "Protein structure prediction using rosetta," in *Numerical Computer Methods, Part D*, ser. Methods in Enzymology, L. Brand and M. L. Johnson, Eds. Academic Press, 2004, vol. 383, pp. 66–93.

[12] H. Ledoux, "Computing the 3d Voronoi diagram robustly: An easy explanation," in *Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 117–129.

[13] G. Wang and R. L. Dunbrack, "PISCES: a protein sequence culling server," *Bioinformatics*, vol. 19, no. 12, pp. 1589–1591, 2003.

[14] H. Edelsbrunner and P. Koehl, "The weighted-volume derivative of a space-filling diagram," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 5, pp. 2203–2208, 2003.

[15] R. Bryant, H. Edelsbrunner, P. Koehl, and M. Levitt, "The area derivative of a space-filling diagram," *Discrete & Computational Geometry*, vol. 32, pp. 293–308, 2004.

[16] L. Yuanxin and J. Snoeyink, *Combinatorial and Computational Geometry*. New York, NY, USA: Cambridge University Press, 2005, ch. 23, pp. 439–458.

[17] L. Guibas and D. Russel, "An empirical comparison of techniques for updating delaunay triangulations," in *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, ser. SCG '04. New York, NY, USA: ACM, 2004, pp. 170–179.