Algorithms **2015**, *8*, 850-869; doi:10.3390/a8040850

OPEN ACCESS

algorithms

ISSN 1999-4893

www.mdpi.com/journal/algorithms

Article

Automatic Classification of Protein Structure Using the Maximum Contact Map Overlap Metric†

Rumen Andonov [1,*], Hristo Djidjev [2], Gunnar W. Klau [3], Mathilde Le Boudic-Jamin [1] and Inken Wohlers [4,5]

[1] INRIA Rennes-Bretagne Atlantique and University of Rennes 1, Campus de Beaulieu, Rennes Cedex 35042, France; E-Mail: mathilde.le_boudic-jamin@inria.fr

[2] Los Alamos National Laboratory, Los Alamos, NM 87544, USA; E-Mail: djidjev@lanl.gov

[3] Life Sciences, CWI, P.O. Box 94079, GB Amsterdam 1090, The Netherlands; E-Mail: Gunnar.Klau@cwi.nl

[4] Genome Informatics, University of Duisburg-Essen, Essen 45147, Germany

[5] Platform for Genome Analytics, Institutes of Neurogenetics & for Integrative and Experimental Genomics, University of Lübeck, Lübeck 23562, Germany; E-Mail: Inken.Wohlers@uni-luebeck.de

† This paper is an extended version of our paper published in Algorithms for Computational Biology. Wohlers, I.; Le Boudic-Jamin, M.; Djidjev, H.; Klau, G. W.; Andonov, R. Exact Protein Structure Classification Using the Maximum Contact Map Overlap Metric, In the Proceeding of the First International Conference, AlCoB 2014, Tarragona, Spain, 1–3 July 2014; pp.262–273.

* Author to whom correspondence should be addressed; E-Mail: rumen.andonov@irisa.fr; Tel.: +33-299847156; Fax: +33-299847171.

Academic Editor: Giuseppe Lancia

Received: 27 June 2015 / Accepted: 16 September 2015 / Published: 9 October 2015

Abstract: In this work, we propose a new distance measure for comparing two protein structures based on their contact map representations. We show that our novel measure, which we refer to as the maximum contact map overlap (max-CMO) metric, satisfies all properties of a metric on the space of protein representations. Having a metric in that space allows one to avoid pairwise comparisons on the entire database and, thus, to significantly accelerate exploring the protein space compared to no-metric spaces. We show on a gold standard superfamily classification benchmark set of 6759 proteins that our exact k-nearest neighbor (k-NN) scheme classifies up to 224 out of 236 queries correctly and on a larger, extended version of the benchmark with $60,850$ additional structures, up to 1361 out of

1369 queries. Our k-NN classification thus provides a promising approach for the automatic classification of protein structures based on flexible contact map overlap alignments.

Keywords: maximum contact map overlap; protein space metric; k-nearest neighbor classification; superfamily classification; SCOP

1. Introduction

Understanding the functional role and evolutionary relationships of proteins is key to answering many important biological and biomedical questions. Because the function of a protein is determined by its structure and because structural properties are usually conserved throughout evolution, such problems can be better approached if proteins are compared based on their representations as three-dimensional structures rather than as sequences. Databases, such as SCOP (Structural Classification of Proteins) [1] and CATH [2], have been built to organize the space of protein structures.

Both SCOP and CATH, however, are constructed partly based on manual curation, and many of the currently over $90,000$ protein structures in the Protein Databank (PDB) [3] are still unclassified. Moreover, classifying a newly-found structure manually is both expensive in terms of human labor and slow. Therefore, computational methods that can accurately and efficiently complete such classifications will be highly beneficial. Basically, given a query protein structure, the problem is to find its place in a classification hierarchy of structures, for example to predict its family or superfamily in the SCOP database.

One approach to solving that problem is based on having introduced a meaningful distance measure between any two protein structures. Then, the family of a query protein q can be determined by comparing the distances between q and members of candidate families and choosing a family whose members are "closer" to q than members of the other families, where the precise criteria for deciding which family is closer depend on the specific implementation. The key condition and a crucial factor for the quality of the classification result is having an appropriate distance measure between proteins.

Several such distances have been proposed, each having its own advantages. A number of approaches based on a graph-based measure of closeness called contact map overlap (CMO) [4] have been shown to perform well [5–11]. Informally, CMO corresponds to the maximum size of a common subgraph of the two contact map graphs; see the next section for the formal definition. Although CMO is a widely-used measure, none of the CMO-based distance methods suggested so far satisfy the triangle inequality and, hence, introduce a metric on the space of protein representations. Having a metric in that space establishes a structure that allows much faster exploration of the space compared to non-metric spaces. For instance, all previous CMO-based algorithms require pairwise comparisons of the query with the entire database. With the rapid increase of the protein databases, such a strategy will unavoidably create performance problems, even if the individual comparisons are fast. On the other hand, as we show here, the structure introduced in metric spaces can be exploited to significantly reduce the number of needed comparisons for a query and thereby increase the efficiency of the algorithm, without sacrificing the accuracy of the classification.

In this work, we propose a new distance measure for comparing two protein structures based on their contact map representations. We show that our novel measure, which we refer to as the maximum contact map overlap (max-CMO) metric, satisfies all properties of a metric. The advantages of nearest neighbor searching in metric spaces are well described in the literature [12–14]. We use max-CMO in combination with an exact approach for computing the CMO between a pair of proteins in order to classify protein structures accurately and efficiently in practice. Specifically, we classify a protein structure according to the k-nearest neighbors with respect to the max-CMO metric. We demonstrate that one can speed up the total time taken for CMO computations by computing in many cases approximations of CMO in terms of lower-bound upper-bound intervals, without sacrificing accuracy. We point out that our approach solves the classification problem to provable optimality and that we do so without having to compute all alignments to optimality. We show on a small gold standard superfamily classification benchmark set of 6759 proteins that our exact scheme classifies up to 224 out of 236 queries correctly and on a large, extended version of the dataset that contains $67,609$ proteins, even up to 1361 out of 1369. Our k-NN classification thus provides a promising approach for the automatic classification of protein structures based on flexible contact map overlap alignments.

Amongst the other existing (non-CMO) protein structure comparison methods, we are aware of only one exploiting the triangle inequality. This is the so-called scaled Gauss metric (SGM) introduced in [15] and further developed in [16]. As shown in the above papers, their approach is very successful for automatic classification. Note, however, that the SGM metric is alignment-free; distances can be computed by SGM, but then, another alignment method is required to provide the alignments. In contrast, the max-CMO metric is alignment-based and provides alignments consistent with the max-CMO score. Hence, for the purpose of comparison, here, we provide results obtained by TM-align [17], one of the fastest and most accurate alignment-based methods. Note, however, that the scope of this paper is not to examine classification algorithms based on different concepts in order to note similarities and differences, but simply to illustrate that the max-CMO score can provide a reliable, fully-automatic protein structure classification.

2. The Maximum Contact Map Overlap Metric

We focus here on the notions of contact map overlap (CMO) and the related max-CMO distance between protein structures. A contact map describes the structure of a protein P in terms of a simple, undirected graph $G = (V, E)$ with vertex set V and edge set E. The vertices of V are linearly ordered and correspond to the sequence of residues of P. Edges denote residue contacts, that is pairs of residues that are close to each other. More precisely, there is an edge (i, j) between residues i and j iff the Euclidean distance in the protein fold is smaller than a given threshold. The size $|G| := |E|$ of a contact map is the number of its contacts. Given two contact maps $G_1(V, E_1)$ and $G_2(U, E_2)$ for two protein structures, let $I = (i_1, i_2, \ldots, i_m)$ and $J = (j_1, j_2, \ldots, j_m)$ be subsets of V and U, respectively, respecting the linear order. Vertex sets I and J encode an alignment of G_1 and G_2 in the sense that vertex i_1 is aligned to j_1, i_2 to j_2, and so on. In other words, the alignment (I, J) is a one-to-one mapping between the sets V and U. Given an alignment (I, J), a shared contact (or common edge) occurs if both $(i_k, i_l) \in E_1$ and $(j_k, j_l) \in E_2$ exist. We say in this case that the shared contact (i_k, i_l) is activated by the alignment (I, J).

The maximum contact map overlap problem consists of finding an alignment (I^*, J^*) that maximizes the number of shared contacts, and $\mathrm{CMO}(G_1, G_2)$ denotes then this maximum number of shared contacts between the contact maps G_1 and G_2; see Figure 1.

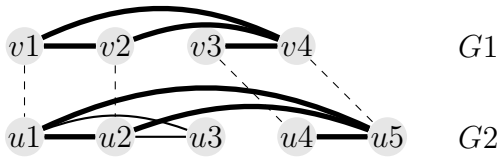

Figure 1. The alignment visualized with dashed lines $((v_1 \leftrightarrow u_1)(v_2 \leftrightarrow u_2)(v_3 \leftrightarrow u_4)(v_4 \leftrightarrow u_5))$ maximizes the number of the common edges between the graphs G_1 and G_2. The four activated common edges are emphasized in bold (*i.e.*, $\mathrm{CMO}(G_1, G_2) = 4$).

Computing $\mathrm{CMO}(G_1, G_2)$ is NP-hard following from [18]. Nevertheless, maximum contact map overlap has been shown to be a meaningful way for comparing two protein structures [5–11]. Previously, several distances have been proposed based on the maximum contact map overlap, for example $D_{\min}$ [5,7] and D_{sum} [6,8,11] with:

$$D_{\min}(G_1, G_2) = 1 - \frac{\mathrm{CMO}(G_1, G_2)}{\min\{|E_1|, |E_2|\}} \text{ and } D_{\mathrm{sum}}(G_1, G_2) = 1 - \frac{2\mathrm{CMO}(G_1, G_2)}{|E_1| + |E_2|}$$

Note that $D_{\min}$ and D_{sum} have been normalized, so that their values are in the interval $[0, 1]$ and are, thus, measures of similarity between proteins. However, they are not metrics, as the next lemma shows.

Lemma 1. *Distances $D_{\min}$ and D_{sum} do not satisfy the triangle inequality.*

Proof. Consider the contact map graphs $G_1, \ldots, G_4$ in Figure 2. It is easily seen that $\mathrm{CMO}(G_1, G_2) = 1, \mathrm{CMO}(G_2, G_3) = 3$ and $\mathrm{CMO}(G_1, G_3) = 3$. We then obtain:

$$D_{\mathrm{sum}}(G_1, G_2) = 1 - \frac{2}{|E_1| + |E_2|} = 1 - \frac{2}{6} = \frac{2}{3}$$

$$D_{\mathrm{sum}}(G_2, G_3) = 1 - \frac{6}{|E_2| + |E_3|} = 1 - \frac{6}{8} = \frac{1}{4}$$

$$D_{\mathrm{sum}}(G_1, G_3) = 1 - \frac{6}{|E_1| + |E_3|} = 1 - \frac{6}{8} = \frac{1}{4}$$

Hence:

$$D_{\mathrm{sum}}(G_1, G_3) + D_{\mathrm{sum}}(G_3, G_2) = \frac{1}{2} < \frac{2}{3} = D_{\mathrm{sum}}(G_1, G_2)$$

Furthermore, $\mathrm{CMO}(G_2, G_4) = 1$ and $\mathrm{CMO}(G_3, G_4) = 2$. We then obtain:

$$D_{\min}(G_2, G_4) = 1 - \frac{\mathrm{CMO}(G_2, G_4)}{\min\{|E_2|, |E_4|\}} = 1 - \frac{1}{3} = \frac{2}{3}$$

and:

$$D_{\min}(G_3, G_4) = 1 - \frac{2}{3} = \frac{1}{3}$$

as well as:

$$D_{\min}(G_2, G_3) = 1 - \frac{3}{3} = 0$$

Hence,

$$D_{\min}(G_2, G_3) + D_{\min}(G_3, G_4) = 0 + \frac{1}{3} < \frac{2}{3} = D_{\min}(G_2, G_4)$$

□

Figure 2. Four contact map graphs.

Let $G_1(V, E_1), G_2(U, E_2)$ be two contact map graphs. We propose a new similarity measure:

$$D_{\max}(G_1, G_2) = 1 - \frac{\mathrm{CMO}(G_1, G_2)}{\max\{|E_1|, |E_2|\}} \tag{1}$$

The following claim states that $D_{\max}$ is a distance (metric) on the space of contact maps, and we refer to it as the max-CMO metric.

Lemma 2. *$D_{\max}$ is a metric on the space of contact maps.*

Proof. To prove the triangle inequality for the function $D_{\max}$, we consider three contact maps $G_1(V, E_1), G_2(U, E_2), G_3(W, E_3)$, and we want to prove that $D_{\max}(G_1, G_2) + D_{\max}(G_2, G_3) \geq D_{\max}(G_1, G_3)$. We will use the fact that a similar function d_{max} on sets is a metric [19], which is defined as:

$$d_{\max}(A, B) = 1 - \frac{|A \cap B|}{\max\{|A|, |B|\}} \tag{2}$$

The mapping $\mathcal{M}$ corresponding to $\mathrm{CMO}(G_1, G_2)$ generates an alignment (V', U'), where $V' \subseteq V$ and $U' \subseteq U$ are ordered sets of vertices preserving the order of V and U, correspondingly. Since $\mathcal{M}$ is a one-to-one mapping, we can rename the vertices of U' to the names of the corresponding vertices of V' and keep the old names of the vertices of $U \setminus U'$. Denote the resulting ordered vertex set by $\overline{U}$, and denote by $\overline{E_2}$ the corresponding set of edges. Define the graph $\overline{G_2} = (\overline{U}, \overline{E_2})$. Note that $|\overline{E_2}| = |E_2|$ and

any common edge discovered by $\mathrm{CMO}(G_1, G_2)$ has the same endpoints (after renaming) in $\overline{E_2}$ as in E_1; hence, $\mathrm{CMO}(G_1, G_2) = \mathrm{CMO}(G_1, \overline{G_2}) = |E_1 \cap \overline{E_2}|$. Then, from Equation (2):

$$D_{\max}(G_1, G_2) = 1 - \frac{\mathrm{CMO}(G_1, G_2)}{\max\{|E_1|, |E_2|\}} = 1 - \frac{|E_1 \cap \overline{E_2}|}{\max\{|E_1|, |\overline{E_2}|\}} = d_{\max}(E_1, \overline{E_2})$$

Similarly, we compute the mapping corresponding to $\mathrm{CMO}(\overline{G_2}, G_3)$ and generate an optimal alignment $(\overline{U'}, W')$. As before, we use the mapping to rename the vertices of W' to the corresponding vertices of $\overline{U'}$ and denote the resulting sets of vertices and edges by $\overline{W}$ and $\overline{E_3}$. Similarly to the above case, it follows that $D_{\max}(G_2, G_3) = d_{\max}(\overline{E_2}, \overline{E_3})$. Combining the last two equalities, we get:

$$\begin{aligned} D_{\max}(G_1, G_2) + D_{\max}(G_2, G_3) &= d_{\max}(E_1, \overline{E_2}) + d_{\max}(\overline{E_2}, \overline{E_3}) \\ &\geq d_{\max}(E_1, \overline{E_3}) \end{aligned} \tag{3}$$

On the other hand, $E_1 \cap \overline{E_3}$ contains only edges jointly activated by the alignments (V', U') and $(\overline{U'}, W')$, and its cardinality is not larger than $\mathrm{CMO}(G_1, G_3)$, which corresponds to the optimal alignment between G_1 and G_3. Hence: $|E_1 \cap \overline{E_3}| \leq \mathrm{CMO}(G_1, G_3)$ and, since $|\overline{E_3}| = |E_3|$:

$$d_{\max}(E_1, \overline{E_3}) = 1 - \frac{|E_1 \cap \overline{E_3}|}{\max\{|E_1|, |\overline{E_3}|\}} \geq 1 - \frac{\mathrm{CMO}(G_1, G_3)}{\max\{|E_1|, |E_3|\}} = D_{\max}(G_1, G_3)$$

Combining the last inequality with Equation (3) proves the triangle inequality for $D_{\max}$. The other two properties of a metric, that $D_{\max}(G_1, G_2) \geq 0$ with equality if and only if $G_1 = G_2$ and $D_{\max}(G_1, G_2) = D_{\max}(G_2, G_1)$, are obviously also true. □

If instead of $\mathrm{CMO}(G_1, G_2)$, one computes lower or upper bounds for its value, replacing those values in Equation (1) produces an upper or lower bound for $D_{\max}$, respectively.

3. Nearest Neighbor Classification of Protein Structures

We suggest to approach the problem of classifying a given query protein structure with respect to a database of target structures based on a majority vote of the k-nearest neighbors in the database. Nearest neighbor classification is a simple and popular machine learning strategy with strong consistency results; see, for example, [20].

An important feature of our approach is that it is based on a metric, and we fully profit from all usual benefits when exploiting the structure introduced by that metric. In addition, we also model each protein family in the database as a ball with a specially-chosen protein from the family as the center, see Section 3.1 for details. This allows one to obtain upper and lower bounds for the max-CMO distance in Section 3.2, which are used to define a new dominance rule we call triangle dominance that proves to be very efficient. Finally, we describe in Section 3.3 how these results can be used in a classification algorithm.

3.1. Finding Family Representatives

In order to minimize the number of targets with which a query has to be compared directly, *i.e.*, via computing an alignment, we designate a representative central structure for each family. Let d denote any metric. Each family $\mathcal{F} \in \mathcal{C}$ can then be characterized by a representative structure $R_{\mathcal{F}}$ and a family radius $r_{\mathcal{F}}$ determined by:

$$R_{\mathcal{F}} = \arg\min_{A \in \mathcal{F}} \max_{B \in \mathcal{F}} d(A, B), \quad r_{\mathcal{F}} = \min_{A \in \mathcal{F}} \max_{B \in \mathcal{F}} d(A, B) \tag{4}$$

In order to find $R_{\mathcal{F}}$ and $r_{\mathcal{F}}$, we compute, during a preprocessing step, all pairwise distances within $\mathcal{F}$. We aim to compute these distances as precisely as possible, using a sufficiently long run time for each pairwise comparison. Since proteins from the same family are structurally similar, the alignment algorithm performs favorably, and we can usually compute intra-family distances optimally.

3.2. Dominance between Target Protein Structures

In order to find the target structures that are closest to a query q, we have to decide for a pair of Targets A and B which one is closer. We call such a relationship between two target structures dominance:

Lemma 3 (Dominance). *Protein A dominates protein B with respect to a query q if and only if $d(q, A) < d(q, B)$.*

In order to conclude that A is closer to q than B, it may not be necessary to know $d(q, A)$ and $d(q, B)$ exactly. It is sufficient that A directly dominates B according to the following rule.

Lemma 4 (Direct dominance). *Protein A dominates protein B with respect to a query q if $\overline{d}(q, A) < \underline{d}(q, B)$, where $\overline{d}(q, A)$ and $\underline{d}(q, B)$ are an upper and lower bound on $d(q, A)$ and $d(q, B)$, respectively.*

Proof. It follows from the inequalities $d(q, A) \leq \overline{d}(q, A) < \underline{d}(q, B) \leq d(q, B)$. □

Given a query q, a target A and the representative $R_{\mathcal{F}}$ of the family $\mathcal{F}$ of A, the triangle inequality provides an upper bound, while the reverse triangle inequality provides respectively a lower bound on the distance from query q to target A:

$$d(q, A) \leq d(q, R_{\mathcal{F}}) + d(R_{\mathcal{F}}, A) \text{ and } d(q, A) \geq |d(q, R_{\mathcal{F}}) - d(R_{\mathcal{F}}, A)| \tag{5}$$

We define the triangle upper (respectively lower) bound as:

$$d^{\triangle}(q, A) = \overline{d}(q, R_{\mathcal{F}}) + \overline{d}(R_{\mathcal{F}}, A) \tag{6}$$

$$d_{\triangledown}(q, A) = \max\{\underline{d}(q, R_{\mathcal{F}}) - \overline{d}(R_{\mathcal{F}}, A), \underline{d}(R_{\mathcal{F}}, A) - \overline{d}(q, R_{\mathcal{F}})\} \tag{7}$$

Lemma 5. $d_{\triangledown}(q, A) \leq d(q, A) \leq d^{\triangle}(q, A)$

Proof. $d_{\triangledown}(q,A) = \max\{\underline{d}(q,R_{\mathcal{F}}) - \overline{d}(R_{\mathcal{F}},A), \underline{d}(R_{\mathcal{F}},A) - \overline{d}(q,R_{\mathcal{F}})\} \leq |d(q,R_{\mathcal{F}}) - d(R_{\mathcal{F}},A)| \leq d(q,A) \leq d(q,R_{\mathcal{F}}) + d(R_{\mathcal{F}},A) \leq \overline{d}(q,R_{\mathcal{F}}) + \overline{d}(R_{\mathcal{F}},A) = d^{\triangle}(q,A)$. □

Using Lemma 5, we derive supplementary sufficient conditions for dominance, which we call indirect dominance.

Lemma 6 (Indirect dominance). *Protein A dominates protein B with respect to query q if $d^{\triangle}(q,A) < d_{\triangledown}(q,B)$.*

Proof. $d(q,A) \overset{\text{Lemma 5}}{\leq} d^{\triangle}(q,A) < d_{\triangledown}(q,B) \overset{\text{Lemma 5}}{\leq} d(q,B)$. □

3.3. Classification Algorithm

k-nearest neighbor classification is a scheme that assigns the query to the class to which most of the k targets belong that are closest to the query. In order to classify, we therefore need to determine the k structures with minimum distance to the query and assign the superfamily to which the majority of the neighbors belong. As seen in the previous section, we can use bounds to decide whether a structure is closer to the query than another structure. This can be generalized to deciding whether or not a structure can be among the k closest structures in the following way. We construct two priority queues, called LB and UB, whose elements are $(t, lb(q,t)))$ and $(t, ub(q,t))$, respectively, where q is the query and t is the target. Here, $lb(q,t)$ (respectively $ub(q,t)$) is any lower (respectively upper) bound on the distance between q and t. In our current implementation, we use $D_{\max}$ as a distance, while lower and upper bounds are $d_{\triangledown}(q,t)$ (respectively $d^{\triangle}(q,t)$) or $\underline{d}(q,t)$ (respectively $\overline{d}(q,t)$) where $\underline{d}(q,t)$ and $\overline{d}(q,t)$ are lower and upper bounds based on Lagrangian relaxation. As explained in [8], these bounds can be polynomially computed by a sub-gradient descent method, where each iteration is solved in $O(n^4)$ time, where n is the number of vertices of the contact map graph. However, when the graph is sparse (which is the case of contact map graphs), the above complexity bound is reduced to $O(n^2)$. The practical convergence of the sub-gradient method is unpredictable, but an experimental analysis performed by the authors of [8] suggests that 500 iterations is a reasonable average estimation. The quality of the bounds $\underline{d}(q,t)$ and $\overline{d}(q,t)$ for the purpose of protein classification has been already demonstrated in [9,11,21].

The priority queues LB and UB are sorted in the order of increasing distance. The k-th element in queue UB is denoted by t_k^{UB}. Its distance to the query, $d(q,t_k^{\mathrm{UB}})$, is the distance for which at least k target elements are closer to the query. Therefore, we can safely discard all of those targets that have a lower bound distance of more than $d(q,t_k^{\mathrm{UB}})$ to query q. That is, t_k^{UB} dominates all targets t for which $lb(q,t) > ub(q,t_k^{\mathrm{UB}})$.

We assume that distances between family members are computed optimally (this is actually done in our preprocessing step when computing the family representatives), *i.e.*, $d(A,B) = \underline{d}(A,B) = \bar{d}(A,B)$ if $A, B \in \mathcal{F}$. The algorithm also works if this is not the case, then $d(A,B)$ needs to be replaced by the corresponding Lagrangian bounds at the appropriate places.

4. Experimental Setup

We evaluated the classification performance and efficiency of different types of dominance of our algorithm on domains from SCOPCath [22], a benchmark that consists of a consensus of the two major structural classifications SCOP [1] (Version 1.75) and Cath [2] (Version 3.2.0). We use this consensus benchmark in order to obtain a gold standard classification that very likely reflects structural similarities that are detectable automatically, since two classifications, each using a mix of expert knowledge and automatic methods, agree in their superfamily assignments. For generating SCOPCath, the intersection of SCOP and Cath has been filtered, such that SCOPCath only contains proteins with less than 50% sequence identity. Since this results in a rather small benchmark with only 6759 structures, we added these filtered structures for our evaluation in order to have a much larger, extended version of the benchmark, which is representative of the overlap between the existing classifications SCOP and Cath. There were 264 domains in extended SCOPCath that share more than 50% sequence similarity with a domain in SCOPCath, but do not both belong to the same SCOP family; since their families are perhaps not in SCOPCath and their classification in SCOP and Cath may not agree, we removed them. This way, we obtained $60,850$ additional structures (*i.e.*, the extended benchmark is composed of $67,609$ structures). These belong to 1348 superfamilies and 2480 families, of which 2093 families have more than one member. For SCOPCath, there are 1156 multi-member families. Structures and families are divided into classes according to Table 1. For superfamily assignment, we compared a structure only to structures of the corresponding class, since class membership can in most cases be determined automatically, for example by a program that computes secondary structure content. In rare cases where class membership is unclear, one could combine the target structures of possible classes before classification. The four major protein classes are labeled from a to d and refer to: (a) all α proteins, *i.e.*, consisting of α-helices; (b) all β proteins, *i.e.*, consisting of β-sheets; (c) α and β proteins with parallel β sheets, *i.e.*, β-α-β units; and (d) α and β proteins with antiparallel β sheets, *i.e.*, segregated α and β regions. These classes are thus defined by secondary structure content and arrangement, which, in turn, is defined by class-specific contact map patterns. We therefore consider them individually when characterizing our max-CMO metric.

Table 1. For every protein class, the table lists the number of structures in SCOPCath (str) and extended SCOPCath (ext), the corresponding number of families (fam) and superfamilies (sup).

Class	a	b	c	d	e	f	g	h	i	j	k
# str	1195	1593	1774	1591	30	103	342	72	11	38	10
# ext	10,796	19,215	17,497	15,679	349	1006	2398	520	43	81	25
# fam	524	516	548	632	6	59	121	32	5	29	8
# sup	303	266	191	375	6	52	82	31	5	29	8

For classification, we randomly selected one query from every family with at least six members. This resulted in 236 queries for SCOPCath and 1369 queries for the extended SCOPCath benchmark.

We then computed all-*versus*-all distances, Equation (1), or distance bounds within each family using optimal maximum contact map overlap or the Lagrangian bound on it. For obtaining the latter, we use our Lagrangian solver A_purva [8] (see also https://www.irisa.fr/symbiose/software, as well as http://csa.project.cwi.nl/), which reads PDB files, constructs contact maps and returns (bounds on) the contact map overlap. Using corresponding distance bounds, we determined the family representative according to Equation (4). The complexity of this step is $\sum_{\forall \mathcal{F} \in \mathcal{C}} |\mathcal{F}|^2$, where $|\mathcal{F}|$ denotes the number of the members of the family $\mathcal{F}$. Note that this step is query independent and is performed as a preprocessing. For every pairwise distance computation, we used a maximum time limit of 10 s. Since most comparisons were computed optimally, the average run time is approximately 2 s.

For every query, the $k = 10$ nearest neighbor structures from SCOPCath and extended SCOPCath, respectively, were computed using our k-NN Algorithm 1. The algorithm is a two-step procedure. First, it improves bounds by applying several rounds of triangle dominance, for which the alignment from query to representatives is computed, and second, it switches to pairwise dominance, for which the alignment to any remaining target is computed. In the first step, query representative alignments are computed using an initial time limit of $\tau = 1$ s; then, triangle dominance is applied to all targets, and the algorithm iterates with the time limit doubled until a termination criterion is met. This way, bounds on query target distances are improved successively. Since the query is compared uniquely with the family representative, only $\sum_{\forall \mathcal{F} \in \mathcal{C}} 1$ alignments are needed at each iteration. The computation of triangle dominance terminates if any of the following holds: (i) k targets are left; (ii) all query-representative distances have been computed optimally or with a time limit of 32 CPU seconds; (iii) the number of targets did not reduce from one round to the next. Pairwise dominance terminates if any of the following holds: (i) k targets are left; (ii) all remaining targets belong to the same superfamily; (iii) all query-target distances have been computed with a time limit of 32 CPU seconds. The query is then assigned to the superfamily to which the majority of the k-nearest neighbors belongs. In cases in which the pairwise dominance terminates with more than k targets or more than one superfamily remains, the exact k-nearest neighbors are not known. In that case, we order the targets based on the upper bound distance to the query and assign the superfamily using the top ten queries. In the case that there is a tie among the superfamilies to which the top ten targets belong, we report this situation.

We compare our exact k-NN classifier with respect to classification accuracy with k-NN classification using TM-align [17] (Version 20130511). TM-align is a widely-used, fast structure alignment heuristic, which the authors, amongst others, applied for fold classification. TM-align alignments further were shown to have high accuracy with respect to manually-curated reference alignments [23,24]. Using TM-align, we align each query to all targets of the same class and compute the corresponding TM-score. The targets are then ranked based on TM-score (normalized with respect to query), and the superfamily that most of the k nearest neighbors belong to is assigned.

In order to investigate the impact of k on classification accuracy, we additionally decreased k from nine to one, using each time the $k+1$ nearest neighbors from the classification result for $k+1$. In the case that for a query, more than $k+1$ targets remained in this classification, we used all of them for searching for the k-nearest neighbors, but put an additional termination criterion if the number of structures after two or more iterations of pairwise dominance exceeds a given number. This effects only about a dozen

queries that needed an extremely long run time for $k = 10$. If this termination criterion is applied, we do not obtain an exact classification, but shorter run times.

Algorithm 1 Solving the k-NN classification problem

q // Query structure.
$\mathcal{T}$ // Set of target structures.
$R_F \; \forall F \in \mathcal{C}$ // Family representatives; see Equation (4).
$d(A, R_F) \; \forall A \in \mathcal{F}$ for all families $\mathcal{F} \in \mathcal{C}$ // Distance from all family members to the respective representative.
$\underline{d}(q, R_F), \quad \overline{d}(q, R_F) \quad \forall F \in \mathcal{C}$ // Bounds on the distance from the query to the family representatives.
LB $\leftarrow \{(t, -\infty) | t \in \mathcal{T}\}$ // Priority queue, which will hold the targets t in the order of increasing lower bound distance $d_{\triangledown}(q, t)$ to the query.
UB $\leftarrow \{(t, \infty) | t \in \mathcal{T}\}$ // Priority queue, which will hold the targets t in the order of increasing upper bound distance $d^{\triangle}(q, t)$ to the query.
t_k^{UB} // A pointer to the k-th element in UB
$\tau \leftarrow 1$ s // Time limit for pairwise alignment.
for $\mathcal{F} \in \mathcal{C}$ **do**
    FAM$[\mathcal{F}] \leftarrow |\{t \in \mathcal{T} : t$ belongs to family $\mathcal{F}\}|$ // Number of family members.
end for
while $\exists R_{\mathcal{F}} : \underline{d}(q, R_{\mathcal{F}}) \neq \overline{d}(q, R_{\mathcal{F}})$ and $|\mathcal{T}|$ changes **do**
    $\tau \leftarrow \tau \times 2$
    for $\mathcal{F} \in \mathcal{C}$ with FAM$[\mathcal{F}] > 0$ **do**
        Recompute $\underline{d}(q, R_F)$ and $\overline{d}(q, R_F)$ using time limit τ
        for $t \in \mathcal{F}$ **do**
            Update priority of t in LB to $d_{\triangledown}(q, t) = |\underline{d}(q, R_F) - d(R_F, t)|$. // Bound from inverse triangle inequality Equation (7).
            Update priority of t in UB to $d^{\triangle}(q, t) = \overline{d}(q, R_F) + d(R_F, t)$. // Bound from triangle inequality Equation (6).
        end for
    end for
    // Check for targets dominated by t_k^{UB}.
    for target t in $\mathcal{T}$ **do**
        if $d_{\triangledown}(q, t) > d^{\triangle}(q, t_k^{\mathrm{UB}})$ **then**
            $\mathcal{T} \leftarrow \mathcal{T} \setminus t$
            LB $\leftarrow$ LB$\setminus t$
            UB $\leftarrow$ UB$\setminus t$
            FAM$[\mathcal{F}] \leftarrow$ FAM$[\mathcal{F}] - 1$ where $\mathcal{F}$ is the family of t.
        end if
    end for
    if $|\mathcal{T}| = k$ **then**
        return The majority superfamily membership $\mathcal{S}$ among $\mathcal{T}$.
    end if
end while
Apply the dominance protocol for query q and targets $t \in \mathcal{T}$ as described in [9]. (The quality of the bounds $\underline{d}(q, t)$ and $\overline{d}(q, t)$ are improved by stepwise incrementing τ within the given time limit. At each step, the direct dominance (Lemma (4)) is applied for the targets from the updated $\mathcal{T}$.)

5. Computational Results

5.1. Characterizing the Distance Measure

In the first preprocessing step, we evaluate how well our distance metric captures known similarities and differences between protein structures by computing intra-family and inter-family distances. A good distance for structure comparison should pool similar structures, *i.e.*, from the same family, whereas it should locate dissimilar structures from different families far apart from each other. In order to quantify such characteristics, we compute for each family with at least two members a central, representative structure according to Equation (4). Therefore, we compute the distance between any two structures that belong to the same family. Such intra-family distances should ideally be small. We observe that the distribution of intra-family distances differ between classes and are usually smaller than 0.5, except for class c. For the four major protein classes a to d, there is a distance peak close to zero and another one around 0.2. For the four major protein classes, they are visualized in Figure 3.

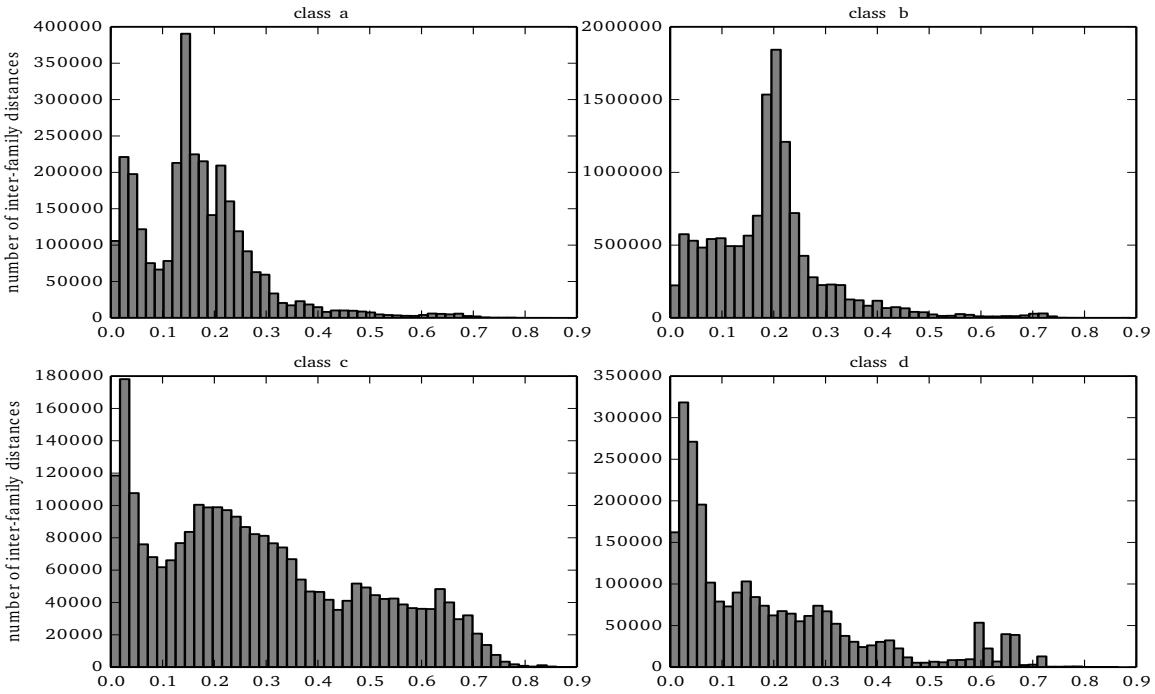

Figure 3. Histograms of intra-family distances divided by class: (**a**) corresponds to class a; (**b**) corresponds to class b; (**c**) corresponds to class c; (**d**) corresponds to class d.

We then compute a radius around the representative structure that encompasses all structures of the corresponding family. The number of families with a given radius decreases nearly linearly from zero to 0.6, with most families having a radius close to zero and almost no families having a radius greater than 0.6. The histogram of family radii is visualized in Figure 4.

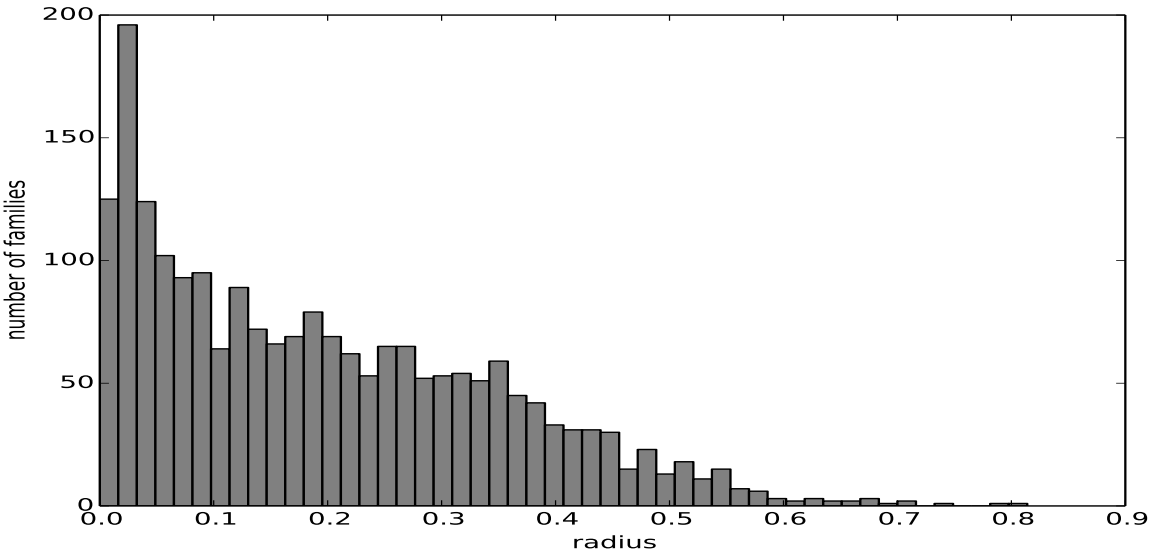

Figure 4. A histogram of the radii of the multi-member families.

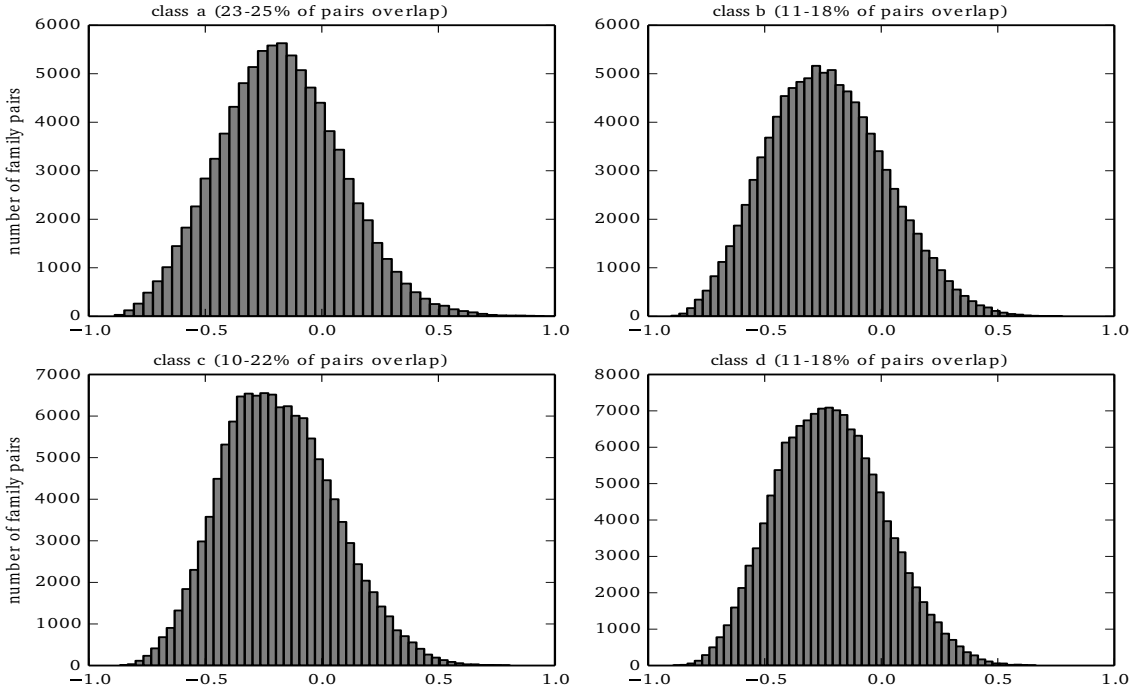

Figure 5. Histograms of overlap values between any two multi-member families for the four main classes a–d: (**a**) corresponds to class a; (**b**) corresponds to class b; (**c**) corresponds to class c; (**d**) corresponds to class d. The title gives an interval on the percentage of overlapping families, computed by using lower and upper bounds, respectively.

Considering that the distance metric is bound to be within zero and one, intra-family distances and radii show that the distance overall captures the similarity between structures well. Further,

we investigate the distance between protein families by computing their overlap value as defined by $r_{\mathcal{F}_1} + r_{\mathcal{F}_2} - d(R_{\mathcal{F}_1}, R_{\mathcal{F}_2})$; for a histogram, see Figure 5. Most families are not close to each other according to our distance metric. Families of the four most populated classes, which belong to different superfamilies, overlap in 23% to 25% of cases for class a, 11% to 18% for class b, 10% to 22% for class c and 11% to 18% for class d. These bounds on the number of overlapping families can be obtained by using the lower and upper bounds on the distances between representatives and the distances between family members appropriately.

5.2. Results for SCOPCath Benchmark

When classifying the 236 queries of SCOPCath, we achieve between 89% and 95% correct superfamily assignments; see Table 2. Remarkably, the highest accuracy is reached for k = 1, so here, just classifying the query as belonging to the superfamily of the nearest neighbor is the best choice. Our k-NN classification resulted for any k in a large number of ties, especially for k = 2; see Table 2. These currently unresolved ties also decrease assignment accuracy compared to $k = 1$, for which a tie is not possible. Table 2 further lists the number of queries that have been assigned, where exact denotes that the provable k nearest neighbors have been computed. The percentage of exactly-computed nearest neighbors varies between 50% and 99% and increases with decreasing k. A likely reason for this is that the larger the k, the weaker is the k-th distance upper bound that is used for domination, especially if the target on rank k is dissimilar to the query. Since SCOPCath domains have low sequence similarity, this is likely to happen. It is also interesting to note that there are for any k quite a few queries that have been assigned exactly, but that are nonetheless wrongly assigned; see Table 2. These are cases in which our distance metric fails in ranking the targets correctly with respect to the gold standard.

Table 2. Classification results showing the number of queries out of the overall 236 queries that have been assigned to a superfamily, the number of correct assignments, the number of assignments computed exactly, thereof the number of correct classifications and the number of ties that do not allow a superfamily assignment based on majority vote. The last two lines display the number of correct assignments and ties for k-NN classification using TM-align.

k	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
# correct	210	211	213	213	214	217	217	219	213	224
# exact	117	143	156	165	188	206	204	211	209	234
# exact and correct	110	134	149	155	178	198	195	205	206	224
# ties	10	9	11	8	10	10	10	10	20	0
# TM-align correct	219	220	220	225	225	228	226	227	226	228
# TM-align ties	4	4	9	5	5	3	8	5	8	0

Figure 6 displays the progress of our algorithm in terms of the percentages of removed targets. We initially compute six rounds of triangle dominance, starting with one CPU second for every query representative alignment and doubling the run time every iteration up to 32 CPU seconds. The same is done in the pairwise dominance step of the algorithm, in which we compute the distance from the query to every remaining target. As shown in Figure 6, the percentage of dominated targets within each

iteration varies widely between queries, which results in a large variance of run times between queries. For some queries, up to 80% of targets can be removed by just computing the distance to the family representatives using a time limit of 1 s and applying triangle dominance; for others, even after several iterations of pairwise dominance, 50% of targets remain. Overall, most queries need, after triangle dominance, several iterations of pairwise dominance before being assigned, and quite a few cannot even be assigned exactly.

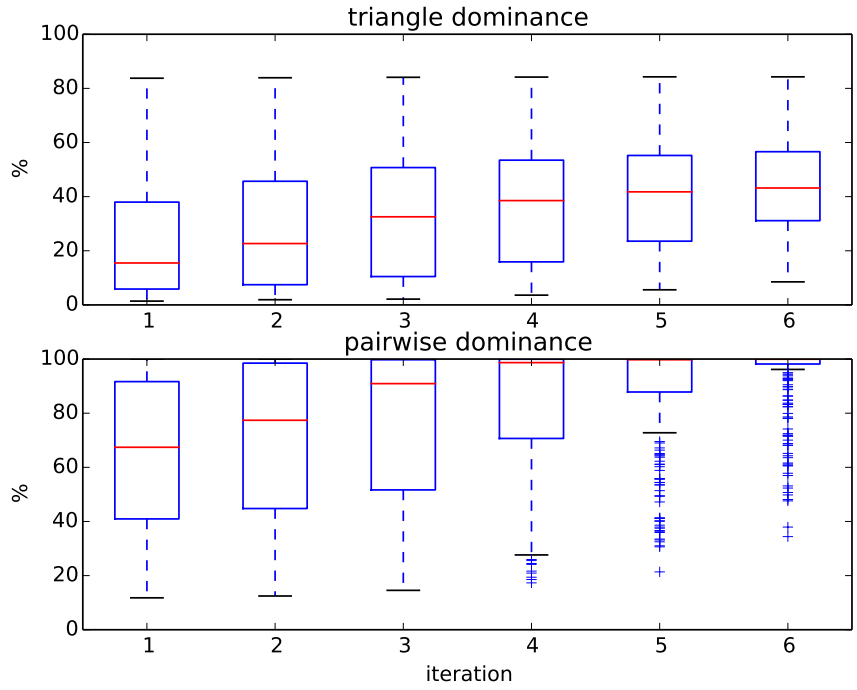

Figure 6. Boxplots of the percentage of removed targets at each iteration during triangle and pairwise dominance for the 236 queries of the SCOPCath benchmark.

5.3. Results for Extended SCOPCath Benchmark

Our exact k-NN classification can also be successfully applied to larger benchmarks, like extended SCOPCath, which are more representative of databases, such as SCOP. Here, the benefit of using a metric distance, triangle inequality and k-NN classification is more pronounced. Remarkably, our classification run time on this benchmark that is about an order of magnitude larger than SCOPCath is for most queries of the same order of magnitude as run times on SCOPCath (except for some queries that need an extremely long run time and finally cannot be assigned exactly). Furthermore, here, run time varies extremely between queries, between 0.15 and 85.63 h for queries of the four major classes that could be assigned exactly. The median run time for all 1120 exactly assigned extended SCOPCath queries is 3.8 h. The classification results for extended SCOPCath are shown in Table 3. Slightly more queries have been assigned correctly compared to SCOPCath, and significantly more queries have been assigned exactly. Both may reflect that there are now more similar structures within the targets. Further, the number of ties is decreased.

Table 3. Classification results showing the number of queries out of the overall 1369 queries that have been assigned to a superfamily, the number of correct assignments, the number of assignments computed exactly, thereof the number of correct classifications and the number of ties that do not allow a superfamily assignment based on majority vote. The last two lines display the number of correct assignments and ties for k-NN classification using TM-align.

k	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
# correct	1303	1331	1334	1341	1341	1346	1344	1351	1348	1361
# exact	1120	1182	1228	1271	1286	1339	1341	1352	1347	1368
# exact and correct	1104	1166	1215	1257	1276	1329	1330	1341	1343	1360
# ties	35	5	12	6	11	7	9	3	17	0
# TM-align correct	1311	1347	1346	1350	1351	1354	1352	1353	1351	1361
# TM-align ties	39	4	7	4	6	4	4	5	15	0

Figure 7 displays the progress of the computation. Here, many more target structures are removed by triangle dominance and within the very first iteration of pairwise dominance compared to the SCOPCath benchmark. For example, for most queries, more than 60% of targets are removed by triangle dominance alone. Only very few queries need to explicitly compute the distance to a large percentage of the targets, and almost 75% of queries can be assigned after only one round of pairwise dominance.

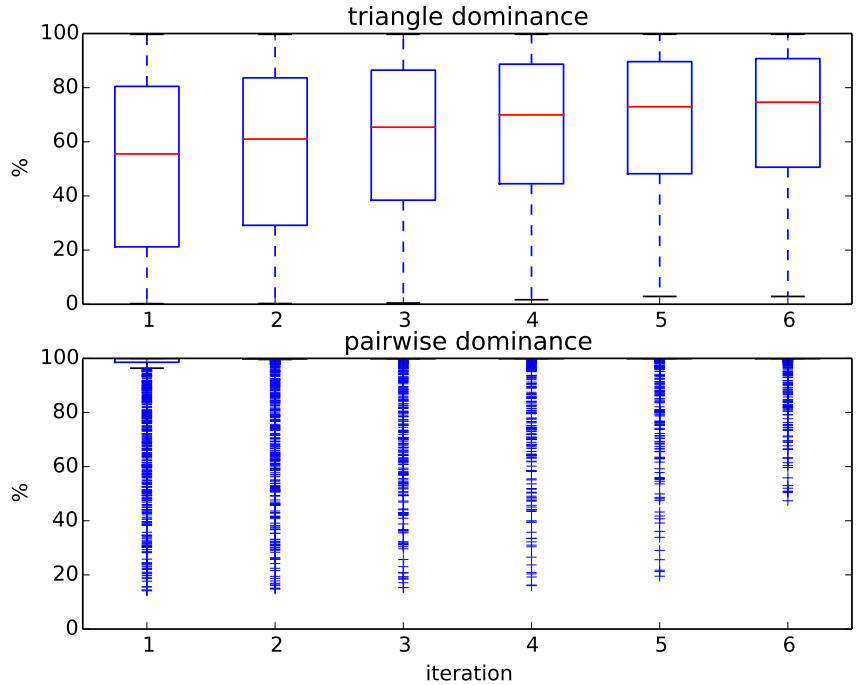

Figure 7. Boxplots of the percentage of removed targets at each iteration during triangle and pairwise dominance for the 1369 queries of the extended SCOPCath benchmark.

6. Discussion

The difficulty to optimally compute a superfamily assignment using k-NN increases with the dissimilarity of the k-th closest target and the query, because this target determines the domination bound, and this bound becomes weaker when k increases. This can be observed in the different number of exactly-assigned queries between SCOPCath and extended SCOPCath, on the one hand, and for different k, on the other hand. Since SCOPCath has been filtered for low sequence identity, we can expect that the k-th neighbor is less similar to the query than the k-th neighbor in extended SCOPCath, and therefore, it is easier to compute extended SCOPCath exactly. Accordingly, the number of exactly-assigned queries tends to increase with decreasing k. In future work, we may use such properties of the distance bounds to decide which k is most appropriate for a given query.

Our exact classification is based on a well-known property of exact CMO computation: similar structures are quick to align and are usually computed exactly, whereas dissimilar structures are extremely slow to align and usually not exactly. Therefore, we remove dissimilar structures early using bounds. Distances between similar structures can then be computed (near-)optimal, and the resulting k-NN classification is exact.

Except for the case $k = 1$ on the extended benchmark, in terms of assignment accuracy, TM-align performs slightly better than max-CMO, and it usually has to some extent fewer ties. On the other hand, both max-CMO and TM-align perform best in the case $k = 1$, and for that most relevant case, the two methods have the same accuracy. Considering that max-CMO is a metric and, thus, needs to compare structures globally, while TM-align is not, it still allows one to perform very accurate superfamily assignment.

While for the extended benchmark, max-CMO and TM-align have the same number of correct classifications for the best choice of value for k, the somewhat better performance of TM-align in the other cases indicates that the max-CMO method could be further improved. A possible disadvantage of our metric is that it does not apply proper length normalization. For instance, if a protein structure is identical to a substructure of another protein, the corresponding max-CMO distance depends on the length of the longer protein. For classification purposes, it would usually be better to rank a protein with such local similarity higher than another protein that is less similar, but of smaller length.

Moreover, although the current results suggest that, in terms of assignment accuracy, using only the nearest neighbor for classification works best, finding the k-nearest neighbor structures is still interesting and important. A new query structure is in need of being characterized, and the set of k closest structures from a given classification gives a useful description on its location in structure space, especially if this space is metric. Note that, besides using the presented algorithm for determining the k-nearest neighbors, it could straightforwardly also be used to find all structures within a certain distance threshold of a given query.

We show that our approach is beneficial for handling large datasets, the structures of which form clusters in some metric space, because it can quickly discard dissimilar structures using metric properties, such as triangle inequality. This way, the target dataset does not need to be reduced previously using a different distance measure, such as sequence similarity, which can lead to mistakes. Our classification is at all times based exclusively on structural distance.

Among the disadvantages of a heuristic approach for the task of large-scale structure classification, we can point to the observation that the obtained classifications are not stable. As versions of tools or random seeds change, the distance between structures may change, since the provable distance between two structures is not known. With these distance changes, also the entire classification may change. Such possible, unpredictable changes in classification contradict the essential use of an automatic classification as a reference. Furthermore, even if a given heuristic could be very fast, it always requires a pairwise number of comparisons for solving the classification problem by the k-NN approach. This requirement obviously becomes a notable hindrance with the natural and quick increase of the protein databases size.

7. Conclusion

In this work, we introduced a new distance based on the CMO measure and proved that it is a true metric, which we call the max-CMO metric. We analyzed the potential of max-CMO for solving the k-NN problem efficiently and exactly and built on that basis a protein superfamily classification algorithm. Depending on the values of k, our accuracy varies between 89% for $k = 10$ and 95% for $k = 1$ for SCOPCath and between 95% and 99% for extended SCOPCath. The fact that the accuracy is highest for $k = 1$ indicates that using more sophisticated rules than k-NN may produce even better results.

In summary, our approach provides a general solution to k-NN classification based on a computationally-intractable measure for which upper and lower bounds are polynomially available. By its application to a gold standard protein structure classification benchmark, we demonstrate that it can successfully be applied for fully-automatic and reliable large-scale protein superfamily classification. One of the biggest advantages of our approach is that it permits one to describe the protein space in terms of clusters with their representative central structures, radii, intra-cluster and inter-clusters distances. Such a formal description is by itself a source of knowledge and a base for future analysis.

Acknowledgments

We are grateful to Noël Malod-Dognin and Nicola Yanev for discussions and useful suggestions and to Sven Rahmann for providing computational infrastructure. We thank the reviewers for a careful reading and for the comments on this study.

Author Contributions

Rumen Andonov, Gunnar W. Klau, Mathilde Le Boudic-Jamin and Inken Wohlers conceived the k-nn classification approach using triangle bounds and the max-CMO metric. Rumen Andonov, Hristo Djidjev and Gunnar W. Klau proved that max-CMO is a metric and other previously used measures not. Inken Wohlers implemented the classification algorithm. Mathilde Le Boudic-Jamin and Inken Wohlers conducted the computational experiments and prepared the results. The authors jointly examined the results and wrote the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Murzin, A.G.; Brenner, S.E.; Hubbard, T.; Chothia, C. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **1995**, *247*, 536–540.
2. Orengo, C.A.; Michie, A.D.; Jones, S.; Jones, D.T.; Swindells, M.B.; Thornton, J.M. CATH—A hierarchic classification of protein domain structures. *Structure* **1997**, *5*, 1093–1108.
3. Bernstein, F.; Koetzle, T.; Williams, G.; Meyer, E.M., Jr.; Brice, M.; Rodgers, J.; Kennard, O.; Shimanouchi, T.; Tasumi, M. The protein data bank: A computer-based archival file for macromolecular structures. *Arch. Biochem. Biophys.* **1978**, *185*, 584–591.
4. Godzik, A.; Skolnick, J.; Kolinski, A. Regularities in interaction patterns of globular proteins. *Protein Eng.* **1993**, *6*, 801–810.
5. Caprara, A.; Carr, R.; Istrail, S.; Lancia, G.; Walenz, B. 1001 optimal PDB structure alignments: Integer programming methods for finding the maximum contact map overlap. *J. Comput. Biol.* **2004**, *11*, 27–52.
6. Xie, W.; Sahinidis, N.V. A reduction-based exact algorithm for the contact map overlap problem. *J. Comput. Biol.* **2007**, *14*, 637–654.
7. Pelta, D.A.; González, J.R.; Vega, M.M. A simple and fast heuristic for protein structure comparison. *BMC Bioinform.* **2008**, *9*, 161.
8. Andonov, R.; Malod-Dognin, N.; Yanev, N. Maximum contact map overlap revisited. *J. Comput. Biol.* **2011**, *18*, 27–41.
9. Malod-Dognin, N.; Boudic-Jamin, M.L.; Kamath, P.; Andonov, R. Using dominances for solving the protein family identification problem. In *Algorithms in Bioinformatics*; Springer: Berlin, Germany, 2011; pp. 201–212.
10. Wohlers, I.; Malod-Dognin, N.; Andonov, R.; Klau, G.W. CSA: Comprehensive comparison of pairwise protein structure alignments. *Nucleic Acids Res.* **2012**, *40*, W303–W309.
11. Malod-Dognin, N.; Przulj, N. GR-Align: Fast and flexible alignment of protein 3D structures using graphlet degree similarity. *Bioinformatics* **2014**, *30*, 1259–1265.
12. Clarkson, K.L. Nearest-neighbor searching and metric space dimensions. *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*; MIT Press: Cambridge, MA, USA, 2006.
13. Moreno-Seco, F.; Mico, L.; Oncina, J. A modification of the LAESA algorithm for approximated k-NN classification. *Pattern Recognit. Lett.* **2003**, *24*, 47–53.
14. Mico, M.L.; Oncina, J.; Vidal, E. A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. *Pattern Recognit. Lett.* **1994**, *15*, 9–17.
15. Rogen, P.; Fain, B. Automatic classification of protein structure by using Gauss integrals. *Proc. Natl. Acad. Sci.* **2003**, *100*, 119–124.
16. Harder, T.; Borg, M.; Boomsma, W.; Røgen, P.; Hamelryck, T. Fast large-scale clustering of protein structures using Gauss integrals. *Bioinformatics* **2012**, *28*, 510–515.

17. Zhang, Y.; Skolnick, J. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* **2005**, *33*, 2302–2309.
18. Lathrop, R.H. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng.* **1994**, *7*, 1059–1068.
19. Horadam, K.; Nyblom, M. Distances between sets based on set commonality. *Discret. Appl. Math.* **2014**, *167*, 310–314.
20. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **1992**, *46*, 175–185.
21. Mavridis, L.; Venkatraman, V.; Ritchie, D.W.; Morikawa, N.; Andonov, R.; Cornu, A.; Malod-Dognin, N.; Nicolas, J.; Temerinac-Ott, M.; Reisert, M.; *et al. SHREC'10 Track: Protein Model Classification*; The Eurographics Association: Norrköping, Sweden, 2010.
22. Csaba, G.; Birzele, F.; Zimmer, R. Systematic comparison of SCOP and CATH: A new gold standard for protein structure analysis. *BMC Struct. Biol.* **2009**, *9*, 23.
23. Wohlers, I.; Domingues, F.S.; Klau, G.W. Towards optimal alignment of protein structure distance matrices. *Bioinformatics* **2010**, *26*, 2273–2280.
24. Wang, S.; Ma, J.; Peng, J.; Xu, J. Protein structure alignment beyond spatial proximity. *Sci. Rep.* **2013**, *3*, 1448.