Exact Protein Structure Classification Using the Maximum Contact Map Overlap Metric

Inken Wohlers, Mathilde Le Boudic-Jamin, Hristo Djidjev, Gunnar W. Klau, Rumen Andonov

Exact protein structure classification using the maximum contact map overlap metric

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

[1] Genome Informatics, University of Duisburg-Essen, Germany, inken.wohlers@uni-due.de
[2] INRIA Rennes - Bretagne Atlantique and University of Rennes 1, France, {rumen.andonov,mathilde.le_boudic-jamin}@irisa.fr
[3] Los Alamos National Laboratory, Los Alamos, NM, USA, djidjev@lanl.gov
[4] Life Sciences, CWI, Science Park 123, 1098 XG Amsterdam, The Netherlands, gunnar.klau@cwi.nl

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 to avoid pairwise comparisons on the entire database and thus to significantly accelerate exploring the protein space compared to non-metric spaces. We show on a gold-standard classification benchmark set of $6,759$ and $67,609$ proteins, resp., that our exact k-nearest neighbor scheme classifies up to 95% and 99% of queries correctly. Our k-NN classification thus provides a promising approach for the automatic classification of protein structures based on contact map overlap.

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 [13] and CATH [14] 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 $98,000$ protein structures in the protein data bank (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. Recently, a number of approaches based on a graph-based measure of closeness called *contact map overlap (CMO)* [7] have been shown to perform well [2, 5, 10, 11, 15, 18, 19]. 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 satisfies the triangle inequality and, hence, introduces 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.

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. This enables us to describe a given protein database as a metric space where we model each protein family as a ball with a specially chosen protein from the family as center. We exploit this representation to accurately and efficiently classify a query protein according to its k nearest neighbors. We demonstrate that using polynomial-time approximations of max-CMO in terms of lower-bound upper-bound intervals speeds up the classification process significantly, without sacrificing its accuracy. We point out that our approach is not heuristic and guarantees solving the classification problem to *provable optimality* with respect to our max-CMO metric and that we do so without having to compute all query-target alignments to optimality.

The metric property has been proved for a related graph distance based on the maximum common subgraph of two graphs [4]. Here, however, we consider contact map overlap, which is based on an underlying order-preserving alignment. Furthermore, the size of a contact map overlap is the number of common edges, whereas the graph distance is based on the number of common nodes.

We show on a gold-standard classification benchmark set of $6,759$ proteins that our exact k-nearest neighbor scheme classifies up to 224 out of 236 queries correctly, and on a large, extended version of the data set that contains $67,609$ proteins even 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.

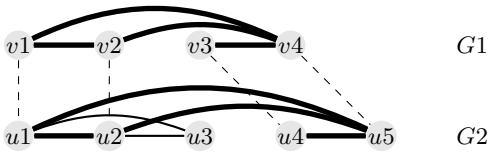

Fig. 1. The alignment visualized with dashed lines ($(v_1 \leftrightarrow u_1)(v_2 \leftrightarrow u_2)(v_3 \leftrightarrow u_4)(v_4 \leftrightarrow u_5)$) maximizes the number of the common edges between the graphs G_1 and G_2. The alignment activates four common edges that are emphasized in bold (i.e.,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

Amongst the other existing (non-CMO) protein structure comparison methods we are aware of only one satisfying the triangle inequality. This so called scaled Gauss metric (SGM) introduced in [16] and further developed in [8] is shown to be very successful in automatic classification. In our work, however, we focus on contact map overlap and a comparison to classification algorithms based on different concepts is outside the scope of this paper.

2 The maximum contact map overlap metric

We introduce here 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 overlap problem consists in 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.

Computing $\mathrm{CMO}(G_1, G_2)$ is NP-hard following from [9]. Nevertheless, maximum contact map overlap has been shown to be a meaningful way for comparing two protein structures [2,5,10,11,18,19]. Previously, several distances have been proposed based on the maximum contact map overlap, for example, $D_{\min}$ [5,15]

and D_{sum} [2, 10, 19] with

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

These distances have the disadvantage that they are no metrics as the following lemma shows (see the extended version [17] for a proof).

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

Let $G_1(V, E_1), G_2(U, E_2)$ be two contact maps graphs. We propose a new distance

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

The following claim states that $D_{\max}$ is indeed 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, which is defined as

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

The mapping $\mathcal{M}$ corresponding to $\text{CMO}(G_1, G_2)$ 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 $\text{CMO}(G_1, G_2)$ has the same endpoints (after renaming) in $\overline{E_2}$ as in E_1; hence $\text{CMO}(G_1, G_2) = \text{CMO}(G_1, \overline{G_2}) = |E_1 \cap \overline{E_2}|$. Then from (2)

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

Similarly, we compute the mapping corresponding to $\text{CMO}(\overline{G_2}, G_3)$ 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 (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 (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 [1]. An important feature of our approach is that it is based on a metric and we fully profit from all usual benefits when exploring a metric space [12].

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 precise 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. These distances obtained during preprocessing are later re-used during k-NN classification for computing triangle bounds.

3.2 Dominance between target protein structures

In order to find the target structures which 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*:

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

The idea of dominance is crucial for reducing the number of computations in our approach. Based on the relationship of polynomial-time lower and upper bounds a dominated protein is discarded from further consideration. Although the precise distance between proteins and the associated alignment are not computed, which is an NP-hard problem, the accuracy of the classification is not sacrificed. In its simplest form this idea has been first proposed in [11]. Here we extend it by exploiting the properties of a metric space as shown below.

Given a query q, a representative r and a target 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

$$d(q, A) \leq d(q, r) + d(r, A) \text{ and } d(q, A) \geq |d(q, r) - d(r, A)| \ .$$

We define the *triangle upper (resp. lower) bound* as

$$\overline{d}^{\triangle}(q, A) = \min_{r \in R} \{\overline{d}(q, r) + \overline{d}(r, A)\} \ ,$$

$$\underline{d}_{\triangledown}(q, A) = \max_{r \in R} \max\{\underline{d}(q, r) - \overline{d}(r, A),\, \underline{d}(r, A) - \overline{d}(q, r)\} \ .$$

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

Proof. $\underline{d}_{\triangledown}(q, A) = \max_{r \in R} \max\{\overline{d}(q, r) - \underline{d}(r, A), \overline{d}(r, A) - \underline{d}(q, r)\} \leq \max_{r \in R} |d(q, r) - d(r, A)| \leq d(q, A) \leq \min_{r \in R} d(q, r) + d(r, A) \leq \min_{r \in R} \overline{d}(q, r) + \overline{d}(r, A) = \overline{d}^{\triangle}(q, A)$. □

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

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

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

3.3 Classification algorithm

K-nearest neighbor classification is a scheme which assigns the query to the class to which most of the k targets belong which 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 super-family 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 is it possible for a structure to be among the k closest structures in the following way. We construct two priority queues LB and UB whose elements are $(t, lb(q,t)))$ and $(t, ub(q,t))$, respectively, where q is the query and t the target. The algorithm works with any lower bound $lb(q,t)$ on the distance between q and t, for example $\underline{d}(q,t)$ or $\underline{d}_{\bigtriangledown}(q,t)$ and with any upper bound $ub(q,t)$ on $d(q,t)$, for example $\overline{d}(q,t)$ or $\overline{d}^{\bigtriangleup}(q,t)$. In our current implementation we use $D_{\max}$ as a distance while lower and upper bounds are polynomially computed based on Lagrangian relaxation as explained in [2]. The quality of these bounds for the purpose of protein classification has been already demonstrated in [10,11]. 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 those targets which 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 $lb(q,t) > ub(q, t_k^{\mathrm{UB}})$.

4 Validation setup

We evaluated the classification performance and efficiency of different types of dominance of our algorithm on domains from SCOPCath [6], a benchmark that consists of a consensus of the two major structural classifications SCOP [13] (version 1.75) and Cath [14] (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 super-family assignments. SCOPCath has been filtered such that it only contains proteins with less than 50% sequence identity. Since this results in a rather small benchmark with only $6,759$ structures, we added these filtered structures for our evaluation in order to have a benchmark representative of the merged databases SCOP and Cath. There were 264 domains in extended SCOPCath which 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. These belong to $1,348$ super-families and $2,480$ families of which $2,093$ families have more than one member. For SCOPCath, there are $1,156$ multi-member families. Structures and families are divided into classes according to Table 1. For super-family assignment, we compared a structure only to structures of the corresponding class

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	10796	19215	17497	15679	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

since class membership can in most cases be determined automatically, for example by a program that computes secondary structure content. We then computed all-versus-all distances (2) or distance bounds within each family using optimal maximum contact map overlap or the upper Lagrangian bound on it and determined the family representative according to Equation (4). 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 classification, we randomly selected one query from every family with at least six members. This resulted in 236 queries for SCOPCath and 1,369 queries for the extended SCOPCath benchmark. For every query, the $k = 10$ nearest neighbor structures from SCOPCath and extended SCOPCath, respectively, were computed using our k-NN Algorithm. The algorithm is a two-step procedure. First it improves distance bounds by applying several rounds of triangle dominance, in which the maximum contact map overlap bounds from query to representatives are updated, and second it switches to pairwise dominance, for which the distance to any remaining target is computed. In the first step, query representative distances are computed using an initial time limit of $\tau = 1$ s, then triangle dominance is applied to all targets and the algorithm iterates with time limit doubled until a termination criterion is met. This way, bounds on query target distances are improved successively. 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 or all remaining targets belong to the same super-family (ii) all query-target distances have been computed with a time limit of 32 CPU seconds. The query is then assigned to the super-family 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 super-family 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 super-family 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. In order to investigate the impact of k on classification accuracy, we additionally decreased k from 9 to 1, using each time the $k + 1$ nearest neighbors from the classification result for

$k+1$. In the case that for a query more than $k+1$ queries remained in this classification, we used all of them for searching for the k nearest neighbors, but put an additional termination criterion which prevents extremely long run times for a few queries. Due to the large number of computations, classifications were run on different architectures on clusters with various load and are therefore only used for order of magnitude comparison.

5 Computational results

5.1 Characterizing the distance measure

In a 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, there is a distance peak close to 0 and another one around 0.2.

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 0 to 0.6, with most families having a radius close to zero, and almost no families having a radius greater than 0.6.

Considering that the distance metric is bound to be within 0 and 1, inter-family distances and radii show that our distance overall captures well the similarity between structures. Further, we investigate the distance between protein families by computing their overlap value as defined by $d(R_{\mathcal{F}_1}, R_{\mathcal{F}_2}) - r_{\mathcal{F}_1} - r_{\mathcal{F}_2}$. 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 23-25% of cases for class a, 11-18% for class b, 10-22% for class c and 11-18% 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 the SCOPCath benchmark

When classifying the 236 queries of SCOPCath, we achieve between 89 and 95% correct super-family 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

Table 2. Classification results showing the number of queries out of overall 236 queries for SCOPCath and 1369 queries for extended SCOPCath that have been assigned to a super-family, the number of assignments to the correct superfamily (cor), the number of assignments computed exactly, i.e. queries which terminate with the provable k nearest neighbors (exc), thereof the number of correct classifications (e&c) and the number of ties which do not allow a superfamily assignment based on majority vote.

	SCOPCath				**ext. SCOPCath**			
k	cor	exc	e&c	ties	cor	exc	e&c	ties
10	210	117	110	10	1303	1120	1104	35
9	211	143	134	9	1331	1182	1166	5
8	213	156	149	11	1334	1228	1215	12
7	213	165	155	8	1341	1271	1257	6
6	214	188	178	10	1341	1286	1276	11
5	217	206	198	10	1346	1339	1329	7
4	217	204	195	10	1344	1341	1330	9
3	219	211	205	10	1351	1352	1341	3
2	213	209	206	20	1348	1347	1343	17
1	224	234	224	0	1361	1368	1360	0

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 which 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 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 which have been assigned exact, but which are nonetheless wrongly assigned, see Table 2. These are cases in which our distance metric fails in ranking the targets correctly with respect to gold standard.

5.3 Results for the 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 which need an extremely long run time and finally cannot be assigned exactly). Also here, run time varies extremely between queries, between 0.15 and 85.63 hours for queries of the four major classes which could be assigned exactly. The median run time for all 1120 exactly assigned extended SCOPCath queries is 3.8 hours.

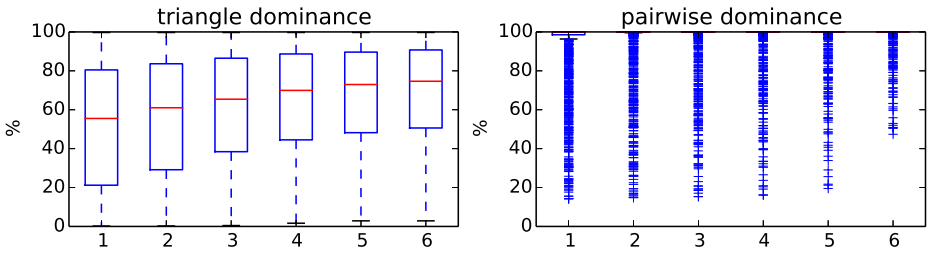

Fig. 2. Boxplots of the percentage of removed targets at each iteration during triangle and pairwise dominance for the 1369 queries of the extended SCOPCath benchmark.

The classification results for extended SCOPCath are shown in Table 2. Slightly more queries have been assigned correctly compared to SCOPCath, between 95 and 99%, 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. Figure 2 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 with 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 most can be assigned after only one round of pairwise dominance.

6 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 value 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 metric for which polynomial upper and lower bounds are available that can successfully be applied for exact large-scale protein superfamily classification.

References

1. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician (1992)
2. Andonov, R., Malod-Dognin, N., Yanev, N.: Maximum contact map overlap revisited. J Comput Biol 18(1), 27–41 (2011)

3. Bernstein, F., Koetzle, T., Williams, G., Jr., E.M., Brice, M., Rodgers, J., Kennard, O., Shimanouchi, T., Tasumi, M.: The protein data bank: A computer-based archival file for macromolecular structures. J. of Mol. Biol. 112, 535 (1977)
4. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters 19, 255–259 (1998)
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 11(1), 27–52 (2004)
6. Csaba, G., Birzele, F., Zimmer, R.: Systematic comparison of SCOP and CATH: a new gold standard for protein structure analysis. BMC Struct Biol 9, 23–23 (2009)
7. Godzik, A., Skolnick, J., Kolinski, A.: Regularities in interaction patterns of globular proteins. Protein Eng 6(8), 801–810 (1993)
8. Harder, T., Borg, M., Boomsma, W., Røgen, P., Hamelryck, T.: Fast large-scale clustering of protein structures using Gauss integrals. Bioinformatics 28(4), 510–515 (2012)
9. Lathrop, R.H.: The protein threading problem with sequence amino acid interaction preferences is NP-complete. Protein Eng 7(9), 1059–1068 (1994)
10. Malod-Dognin, N., Przulj, N.: Gr-align: fast and flexible alignment of protein 3d structures using graphlet degree similarity. Bioinformatics (2014)
11. Malod-Dognin, N., Boudic-Jamin, M.L., Kamath, P., Andonov, R.: Using dominances for solving the protein family identification problem. In: Algorithms in Bioinformatics, Lecture Notes in Computer Science, vol. 6833, pp. 201–212 (2011)
12. Moreno-Seco, F., Mico, L., Oncina, J.: A modification of the laesa algorithm for approximated k-nn classification. Pattern Recognition Letters 24, 47–53 (2003)
13. 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 247(4), 536–540 (1995)
14. 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 5(8), 1093–1108 (1997)
15. Pelta, D.A., González, J.R., Moreno Vega, M.: A simple and fast heuristic for protein structure comparison. BMC Bioinformatics 9, 161–161 (2008)
16. Rogen, P., Fain, B.: Automatic classification of protein structure by using gauss integrals. Proceedings of the National Academy of Sciences of the United States of America 100(1), 119–124 (2003)
17. Wohlers, I., Boudic-Jamin, M.L., Djidjev, H., Klau, G.W., Andonov, R.: Exact protein structure classification using the maximum contact map overlap metric. Tech. Rep. LA-UR-14-20815, Los Alamos National Laboratory (2014)
18. Wohlers, I., Malod-Dognin, N., Andonov, R., Klau, G.W.: CSA: comprehensive comparison of pairwise protein structure alignments. Nucleic Acids Research 40(W1), W303–W309 (2012)
19. Xie, W., Sahinidis, N.V.: A reduction-based exact algorithm for the contact map overlap problem. J Comput Biol 14(5), 637–654 (2007)