BIOINFORMATICS

Vol. 20 no. 7 2004, pages 1015–1021
DOI: 10.1093/bioinformatics/bth031

Measuring the similarity of protein structures by means of the universal similarity metric

N. Krasnogor[1,] and D. A. Pelta[2]*

[1]Automated Scheduling, Optimisation and Planning Group, University of Nottingham, Nottingham, NG8 1BB, UK and [2]Department of Computer Science and Artificial Intelligence, E.T.S.I. Informatica, Universidad de Granada, 18071 Granada, Spain

Received on June 26, 2003; revised on October 13, 2003; accepted on October 22, 2003
Advance Access publication January 29, 2004

ABSTRACT

Motivation: As an increasing number of protein structures become available, the need for algorithms that can quantify the similarity between protein structures increases as well. Thus, the comparison of proteins' structures, and their clustering accordingly to a given similarity measure, is at the core of today's biomedical research. In this paper, we show how an algorithmic information theory inspired Universal Similarity Metric (USM) can be used to calculate similarities between protein pairs. The method, besides being theoretically supported, is surprisingly simple to implement and computationally efficient.

Results: Structural similarity between proteins in four different datasets was measured using the USM. The sample employed represented alpha, beta, alpha–beta, tim–barrel, globins and serpine protein types. The use of the proposed metric allows for a correct measurement of similarity and classification of the proteins in the four datasets.

Availability: All the scripts and programs used for the preparation of this paper are available at http://www.cs.nott.ac.uk/~nxk/USM/protocol.html. In that web-page the reader will find a brief description on how to use the various scripts and programs.

Contact: Natalio.Krasnogor@nottingham.ac.uk; dpelta@ugr.es

Supplementary information: The protein datasets used are collected in http://www.cs.nott.ac.uk/~nxk/USM/datasets.html. The calculated similarity values for the proteins used in this paper can be found in http://www.cs.nott.ac.uk/~nxk/USM/similar.html. The clustering of the dataset based on these similarity values can be found in http://www.cs.nott.ac.uk/~nxk/USM/clustering.html

1 INTRODUCTION

Nowadays researchers who are interested in analysing and understanding proteins' sequences, structures and functions have more than 30 genomes at their fingertips.[1] The comparison of proteins' structures, and their clustering according to similarity, is a fundamental aspect of today's biomedical research.

The comparison of the three-dimensional structures of protein molecules is a challenging problem. The search for effective solution techniques for this problem, is justified because such tools can aid scientists in the development of procedures for drug design, in the identification of new types of protein architecture, in the organization of the known universe of protein structures and can help to discover unexpected evolutionary and functional inter-relations between proteins (Holm and Sander, 1996; Koehl, 2001). There is yet another important role for measures of similarity and clustering algorithms: the evaluation of *ab-initio*, threading or homology modeling structure predictions. Hence, any advancement in structural similarity measures will also impact on structure prediction and its evaluation methodologies (Siew *et al.*, 2000).

Agreement on which is the best similarity measure to use is not forthcoming and a variety of structure comparison methods have been used in classification servers, such as SCOP (Murzin *et al.*, 1995), DALI (Holm and Sander, 1993), LGA (Zemla, 2000, http://PredictionCenter.llnl.gov/local/lga; Zemla *et al.*, 1999), CATH (Orengo *et al.*, 1997) and others. Each method is usually based on a particular biological conception of structural similarity and they generally use different algorithmic strategies. Methodologies based on dynamic programming (Taylor, 1999), comparisons of distance matrices (Holm and Sander, 1993), maximal common sub-graph detection (Artimiuk *et al.*, 1995), geometrical matching (Wu *et al.*, 1998), consensus shapes (Chew and Kedem, 2002) and structures (Leluk *et al.*, 2003) are but a few of the available tools for structural comparison. Most of the existing methods implicitly accept that a suitable scoring function can be defined for which optimum values correspond to the best possible structural match between two proteins. It is implicitly assumed that, based on these optimal matches, similarity between protein structures can be captured.

Approaches based on root-mean-square-distances (e.g. Maiorov and Crippen, 1994; Cohen and Sternberg, 1980) and differences of distance matrices (e.g. Holm and Sander, 1996) sometimes present numerical instabilities; other algorithms

*To whom correspondence should be addressed.

[1]E.g. visit the NCBI site ftp://ncbi.nlm.nih.gov/genbank/genomes.

cannot produce a proper ranking of protein similarities due to an ambiguous definition of the measure. More recent approaches that partially address the problems mentioned above can be found in Lancia *et al.* (2001), Zemla *et al.* (1999) and Hubard (1999).

A more recent approach for structural matching was introduced in Goldman *et al.* (1999) and extended in Lancia *et al.* (2001), Caprara and Lancia (2002), Carr *et al.* (2002) and Krasnogor (2003). This method is based on the maximum alignment (also called overlap) of contact maps. The optimal alignment of contact maps is the only structural matching method for which exact upper and lower bounds can be computed and compared (see references above). However, as the problem of maximizing the overlap between two contact maps was shown to be NP-hard (Goldman, 2000; Goldman *et al.*, 1999) and later (by a different reduction) in Krasnogor (2002, http://www.cs.nott.ac.uk/∼nxk/papers.html) one is still required to resort to approximate algorithms for its solution. Furthermore, the majority of the similarity measures used by these systems are not 'metrics' in the formal sense of the term.[2] A good review of various similarity measures (37 in total) can be found in May (1999). In this paper, we introduce the reader to the 'universal' similarity metric, that can be used to capture every other similarity metric for protein structures. The problem of measuring and clustering together similar protein structures can be decomposed, on the one hand, in developing a similarity assessment methodology and, on the other hand, developing a suitable clustering methodology. In this paper we are only concerned with the first of the two issues, while the clustering aspect will be addressed using a widely available clustering tool-set with the sole purpose of assessing the measure itself. However, the reader should note that the appropriate clustering method is as important as the appropriate measuring method (Koehl, 2001) and that there is a vast literature related to cluster analysis [see, e.g. Gordon (1999) and references therein]. We show here for the first time how the so-called Universal Similarity Metric (USM in the following section) can be used to compare protein structures. The question of which is the clustering method that can better take advantage of this USM is the object of a future paper.

1.1 The Universal Similarity Metric

The USM approximates every possible similarity metric (i.e. those that exist today and those that are yet to be defined). At the heart of the USM, which was introduced in Li *et al.* (2001) and recently refined in Li *et al.* (2003), lies the concept of Kolmogorov complexity. The Kolmogorov complexity $K(\cdot)$ of an object o is defined by the length of the shortest program for a Universal Turing Machine U that is needed to output o, i.e.

$$K(o) = \min\{|P|,\ P \text{ a program and } U(P) = o\}. \quad (1)$$

It can be shown that the Kolmogorov complexity of an object depends only on that object and varies at most up to an additive constant if a different universal Turing machine is chosen as the reference machine (Li and Vitanyi, 1997). The Kolmogorov complexity is an objective measure of the amount of information contained in a given object. A related measure is the conditional Kolmogorov complexity of o_1 given o_2:

$$K(o_1 \mid o_2) = \min\{|P|,\ P \text{ a program and } U(P, o_2) = o_1\}. \quad (2)$$

Equation (2) measures how much information is needed to produce object 1 if we knew object 2.

Furnished with these concepts it is possible to show (Bennett *et al.*, 1998) that the information distance between two objects is equivalent (up to a logarithmic additive term) to:

$$ID(o_1, o_2) = \max\{K(o_1 \mid o_2), K(o_2 \mid o_1)\}. \quad (3)$$

Definition 3 is none other than the Kolmogorov–Chaitin–Solomonof complexity of describing object o_1 given o_2 and describing object o_2 given o_1. Moreover, it can be proved (Bennett *et al.*, 1998) that it is a proper metric. Various works have used measures similar in spirit to the one that appears in Equation (3). Recently Li *et al.* (2001) produced whole mitochondrial sequence philogeny using a related concept while in Bennett *et al.* (2003) the authors showed how to infer chain letter evolutionary histories and how to detect plagiarism in programming assignments by USM. In Varre *et al.* (1998) the authors measured the transformation distance between the genomes of two species by means of comparing their conditional Kolmogorov complexities, and in Grumbach and Tahi (1994) the compression ratio of sequences was used to measure their similarity. Krasnogor (2002) independently derived Equation (3) to measure the relatedness of different runs of evolutionary computation simulations. One drawback of some available measures (e.g. Varre *et al.*, 1998; Grumbach and Tahi, 1994) is that they are not metrics in the formal sense. On the other hand, while the measure introduced in Li *et al.* (2001) is a proper metric it is not a normalized one[3] and hence some unjustified (dis)similarities may be detected. The Universal Similarity Measure [as introduced in Li *et al.* (2003)] is a proper metric, it is universal and also normalized. The metric is formally defined as:

$$d(o_1, o_2) = \frac{\max\{K(o_1 \mid o_2^*), K(o_2 \mid o_1^*)\}}{\max\{K(o_1), K(o_2)\}}, \quad (4)$$

where $o_{1,2}^*$ indicates a shortest program for o_1(or o_2).

[2] A metric is a non-negative symmetric binary function that satisfies the triangle inequality and is zero only if the objects related by the function are one and the same.

[3] See Li *et al.* (2003) for details.

The universality of the USM is paid by non-computability, i.e. Kolmogorov complexity is non-computable but only upper semi-computable. In Sections 2 and 3 we will show how to approximate the Kolmogorov complexity of protein structures. To the authors best knowledge this is the first time this universal metric has been applied to the measurement of protein structures similarity. For mathematical details and proofs about the normalization, universality, etc. of this metric please refer to Li *et al.* (2003).

2 SYSTEMS AND METHODS

In this paper, we apply the recently discovered 'USM' in order to measure the similarity between pairs of protein structures.

All these inter-pair distances are then stored in a similarity matrix. The matrix is then fed into an off-the-shelf clustering methods with outstanding results.

The strength of the method lies not in the particular clustering method or the choice of contact maps (see below) used to represent the structures but rather in that the USM captures all previous metrics. That is, all the similarity measures mentioned in the introductory section of this paper concentrate in one or more aspects of the domain in questions (i.e. proteins topological fingerprints) and build upon these features a 'heuristic' assessment of similarity. In contrast, USM is 'universal' in a mathematical sense, meaning that for any metric and any pair of objects (i.e. protein structures in this paper), within an additive constant, it will coincide with that metric (whether heuristic or not) on those objects (Li *et al.*, 2003). That is, this new metric can be used as a robust measure of similarity in domains where either, there is not enough modelling information, or there is no consensus on what aspects are to be modelled. We used the following protocol to measure similarity between protein structures:

(1) Choose a protein dataset (we used four different datasets that are described in the text).

(2) Extract from each pdb file the first chain (if other than chain 'A' is used from the pdb file this is shown in the text as pdb accession number and a letter, e.g. 1babB).

(3) Produce a contact map for each of the pdb files in the dataset (the contact maps we used have a distance threshold of 6.5 Å and distances are measured from the C_α atoms).

(4) For each pair of protein contact maps c_1, c_2, compute $d(c_1, c_2)$ using Equation (4) to obtain the similarity distance between them. Store all inter-distances in a matrix.

(5) Use an off-the-shelf software to cluster together proteins based on the inter-distances matrix (see text for details).

In step one we employed four different datasets. The first one was based on a family of randomly created protein structures, while the remaining three were based on datasets recently used in the literature for structure comparison purposes.

Random dataset: This dataset contained 40 randomly generated protein structures. The generated set consisted of 500 residues structures to which either alpha-helix, beta-sheet or alpha–beta content was assigned. This was done by randomly allocating bands parallel (for alpha content) or perpendicular (for beta content) to the main diagonal of the associated contact maps. Additionally, the random alpha and beta families were divided in three sub-families corresponding to low, medium or high alpha or beta content. A fourth family of structures with no distinctive features (i.e. totally random) was also present in the dataset. The dataset was thus composed of: random alpha (R500A1_0, R500A1_1, R500A1_2, R500A1_3, R500A1_4, . . . , R500A3_0, R500A3_1, . . . , R500A3_4), random beta (R500B1_0, R500B1_1, R500B1_2, R500B1_3, R500B1_4, . . . , R500B3_0, R500B3_1, . . . , R500B3_4), random alpha–beta (R500AB2_0, R500AB2_1, R500AB2_2, R500AB2_3, R500AB2_4) and random (R500_0, R500_1, R500_2, R500_3, R500_4).[4]

Chew–Kedem dataset: This dataset was used in Chew and Kedem (2002) to assess the quality of a newly proposed method to measure consensus shapes. These are 36 medium size proteins of 5 different families: globins (1eca, 5mbn, 1hlb, 1hlm, 1babA, 1babB, 1ithA, 1mba, 2hbg, 2lhb, 3sdhA, 1ash, 1flp, 1myt, 1lh2, 2vhbA, 2vhb), alpha–beta (1aa9, 1gnp, 6q21, 1ct9, 1qra, 5p21), tim–barrels (6xia, 2mnr, 1chr, 4enl), all beta (1cd8, 1ci5, 1qa9, 1cdb, 1neu, 1qfo, 1hnf) and alpha (1cnp,1jhg). Protein 2vhb was repeated two times (as 2vhb and 2vhbA) in order to check whether the USM detects that the two are identical and induces a cluster where both appear together.

Skolnick dataset: This dataset was used in various recent papers related to structural comparison of proteins (Krasnogor, 2003; Caprara and Lancia, 2002; Carr *et al.*, 2002; Lancia *et al.*, 2001). We selected here only 39 of these proteins: 1b00A, 1dbwA, 1nat, 1ntr, 1qmpA, 1qmpB, 1qmpC, 1qmpD, 1rn1A, 1rn1B, 1rn1C, 4tmyA, 4tmyB, 3chy, 1bawA, 1byoB, 1kdi, 1nin, 1pla, 2b3iA, 2pcy, 2plt, 1amk, 1aw2A, 1b9bA, 1btmA, 1htiA, 1tmhA, 1treA, 1tri, 3ypiA, 8timA, 1ydvA, 1b71A, 1bcfA, 1dpsA, 1fha, 1ier, 1rcd.

Leluk–Konieczny–Roterman dataset: This is a small dataset very recently employed in Leluk *et al.* (2003) to test a new similarity measure based on geometric parameters of polypeptide chains: 1ovaA, 1att, 2achA, 2achA, 2achI, 2antL, 7apiA.

[4] The nomenclature used is as follows: 'R500' stands for 500 residues structures, 'A', 'B', 'AB' stands for alpha, beta or alpha–beta random contents, respectively. The numbers '1', '2', '3' after the 'A', 'B', 'AB' stands for low, medium, high content of the associated features, while '_0', '_1', etc. is an indication of the particular random instance.

The pdb files for the proteins in these datasets contain miscellaneous information that is not related to our studies. For example, fields like 'Author' or 'Remarks' lines are irrelevant for our purposes and if given to the USM engine then it would also be included in the calculations of similarity producing perhaps a clustering based on authorship rather than on topology. There are other, more subtle reasons why we should pre-filter the pdb entries before computing the universal similarity distances for our datasets (e.g. the atoms' spatial coordinate system might be very different between protein pairs). In order to alleviate this we decided to map the information in the pdb file to a contact map (step three in the protocol) and use the contact maps as the objects for which similarities will be computed. The contact map of each protein captures topological information about the structure and hence any non-essential information is left out of the USM engine.

The fourth step of the protocol, i.e. the actual calculations of the similarity between pairs of proteins, is at the core of this paper and is described in detail in the 'Implementation' section.

We use an off-the-shelf clustering method (step five in the protocol) to visually inspect the resulting distance matrices of datasets analysed. We run the clustering server located in http://www2.biology.ualberta.ca/jbrzusto/cluster.php. The web-server executes a combinatorial hierarchical clustering process that begins with each structure in a cluster of its own. When more than one cluster exist then they are combined in a pairwise fashion, i.e. the two closest clusters are combined into a new cluster. Then an inter-cluster distance is calculated between the new cluster and the pre-existing ones. The inter-cluster distance was calculated as the unweighted arithmetic average distance (i.e. USM distance) between a protein structure in one cluster and a protein structure in a second cluster.

3 IMPLEMENTATION

The fourth step in the protocol described in the previous section necessitates the implementation of Equation (4) which is in fact semi-upper computable. In order to use this universal measurement of similarity we need to find suitable estimators for the Kolmogorov complexity of the contact maps. In this paper we followed the methodology used in Li *et al.* (2003) and Cilibrasi *et al.* (2003, http://arxiv.org/archive/cs/0303025) to estimate $K(\cdot)$. Each contact map is represented as a string s and $K(s)$ is approximated by the size (i.e. number of bytes) of the compressed string zip(s), i.e. $K(s) \approx |\text{zip}(s)|$. In Li and Vitanyi (1997) the authors show that algorithmic information is symmetric, hence we can also approximate $K(o_1 \mid o_2)$ by $K(o_1 \cdot o_2) - K(o_2)$ where '$\cdot$' denotes string concatenation and $K(\cdot)$ is estimated as mentioned above. The compression algorithm used was Linux's 'compress' version 4.2.4. Other compression algorithms, e.g. gzip and bzip2, were also tested without significant improvements to the similarity metric and, in the case of the bzip2 compressor with considerable running time slowdown ($\approx$10 times). We computed the similarities for every pair of proteins in the four datasets using the methodology described above. This step of the protocol is $O(n^2)$ where n is the number of protein structures to be assessed (assuming the inter-distance matrix is built from scratch each time). In terms of wall-clock time, using a Pentium 2.4 GHz with 512 Mb of RAM and 40 Gb of disk space, each of the datasets analysed here took around 18 s to approximate $K(o)$ and 700 s to approximate $K(o_1 \mid o_2)$.

As an initial assessment of the practicality of our approach we computed the USM values for the Random dataset. The results are depicted in Figure 1. A visual inspection of the clustering obtained shows that the proposed measure (upon which the clustering was obtained) correctly captures the structural features (i.e. alpha, beta, alpha–beta, totally random content) that are present in the dataset. It is possible to see that the random instances are classified into four groups, i.e. all the random proteins with alpha, beta and alpha–beta features appear in their own cluster while the totally random proteins are collected on a fourth cluster. If we focus our attention within each of the family clusters we can also note that USM is sensible enough to detect differences between random instances with low, medium and strong content of the associated features. As an example of this, consider the cluster representing the structures with random alpha-helix content where it is possible to see that the more helical instances (e.g. R500A3_0, R500A3_1, R500A3_2) are subclustered together as are those instances with medium and low helical content. A similar situation occurs in the beta cluster. We repeated similar random experiments with proteins in the range of 300, 400 and 600 residues with similar results.

The USM values were also computed for the Chew–Kedem, Skolnick and Leluk–Konieczny–Roterman datasets and these matrices are available in our web-site http://www.cs.nott.ac.uk/~nxk/USM/similar.html. In order to test the validity of the USM as a suitable measurement of protein structural similarity we show in Figure 2 the clustering obtained for the Chew–Kedem dataset.

As it can be seen, even a very simple clustering techniques can make good use of the USM. The approach, which requires very little human intervention, was able to distinguish between the four groups of proteins (i.e. globins, tim–barrels, alpha–beta, all beta) and cluster these accordingly.[5] Figure 2 is an almost perfect clustering of the Chew–Kedem proteins. All the alpha–beta, tim–barrel and globins proteins are adequately clustered. Protein 1hnf is clustered with the globins but in a separate branch of the tree. The reason for this is that this protein is a mainly beta protein (it is marked with a * in the figure) but belongs to the immunoglobulin type of proteins. It is surprising that the USM can detect this sort of affinity

[5] The length of the tree branches are not proportional to the distances but rather were set for better visualization in a reduced space.

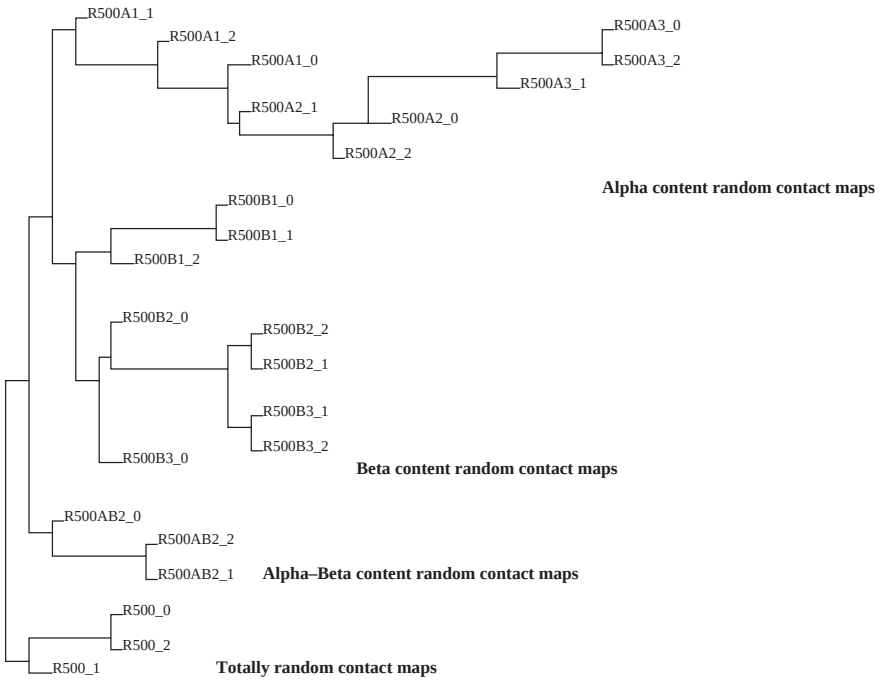

Fig. 1. Clustering of randomly generated families of contact maps according to the Universal Similarity Measure. In the picture, only three out of five random instances for each family are displayed in order to avoid cluttering the graphical representation.

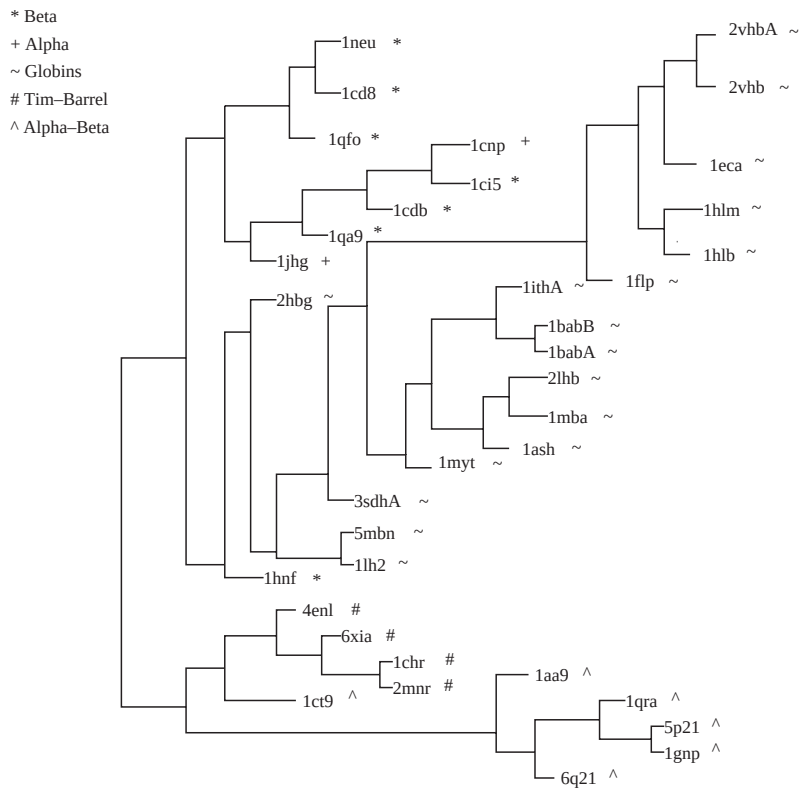

Fig. 2. Clustering of proteins from the Chew–Kedem dataset according to the Universal Similarity Measure.

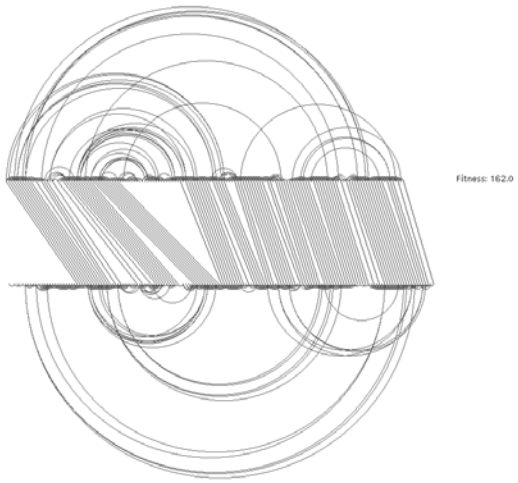

Fig. 3. Equivalent residues for proteins 1ash and 1hlm found by a genetic algorithm under the MAX-CMO model. Red edges are contacts between residues along the protein sequence while blue lines denote equivalent residues across structures.

without human intervention. There are only two (out of 36) proteins that seems to be misplaced, these are 1jhg and 1cnp. They were clustered with the beta proteins when they are actually almost all alpha type of proteins. A closer inspection of their contact maps reveals that, although they mainly contain alpha-helix features, they do possess turns and, to a lesser degree, beta-sheet content. These turns and weak beta features seems to affect the clustering result. An a posteriori maximum contact map overlap (as suggested in the next section) supports this interpretation for their mis-classification. The distance tables and clustering figures for the other two datasets are not shown here (due to space limitations) but are accessible through our web site. Similar results were found with the remaining datasets.

4 DISCUSSION

In previous sections we gave mathematical and experimental evidence that USM can be used to successfully assess protein structures similarity. The USM seems to be capable of capturing protein similarities that encompass a variety of other, more heuristic, criteria in a fully automated way. It seems that the USM is so robust that even with a rough guess of parameters it is still possible to deliver good results. One disadvantage of using USM on its own is that, although it can differentiate between protein families and sub-families and measure similarity based on a rigorous mathematical definition, it does not give indications of where these (di)similarities come from. This drawback can be mitigated by using a two-tier protocol for similarity assessment whereby USM quickly and reliably captures the similarity among protein structures and a maximum contact map overlap obtains a residue equivalence between proteins deemed dissimilar by USM. The computation of the USM was based on a contact map representation not only because contact maps are well suited for capturing topological information but also because they can be used (a posteriori) to obtain residue alignments for the proteins in question. Figure 3 shows (some of) the equivalent residues for proteins 1ash and 1hlm which were classified as similar proteins by USM.

Several research issues merit further investigation: Which is the best type of contact map that should be used in conjunction with USM? What representation is more suitable for the compression algorithm? Which is the best clustering method? Perhaps more importantly, data mining the dictionary that is created during compression of the contact maps could aid in the discovery of unsuspected relationships between protein structures. All these issues are being investigated.

ACKNOWLEDEGEMENTS

The authors would like to acknowledge the anonymous reviewers for their invaluable comments and the time they invested reviewing this paper. D.A.P. is supported by project TIC2002-04242-C03-02.

REFERENCES

Artimiuk,P.J., Poirrette,A.R., Rice,D.W. and Willett,P. (1995) The use of graph theoretical methods for the comparison of the structure of biological macromolecules. *Top. Curr. Chem.*, **174**, 73–103.

Bennett,C.H., Gacs,P., Li,M., Vitanyi,P.M.B. and Zurek,W. (1998) Information distance. *IEEE Trans. Inform. Theor.*, **44**, 1407–1423.

Bennett,C.H., Li,M. and Ma,B. (2003) Chain letters and evolutionary histories. *Sci. Am.*, **288**, 76–81.

Carr,B., Hart,W.E., Krasnogor,N., Burke,E.K., Hirst,J.D. and Smith,J.E. (2002) Alignment of protein structures with a memetic evolutionary algorithm. *GECCO-2002: Proceedings of the Genetic and Evolutionary Computation Conference*. Morgan Kaufman, NY.

Chew,L.P. and Kedem,K. (2002) Finding consensus shape for a protein family. *18th ACM Symposium on Computational Geometry*, ACM Press, Barcelona, Spain.

Caprara,A. and Lancia,G. (2002) Structural alignment of large-size proteins via lagrangian relaxation. *Proceedings of RECOMB 2002*. ACM Press, Washington, DC.

Cohen,F.E. and Sternberg,M.J.E. (1980) On the prediction of protein structure: the significance of the root mean square deviation. *J. Mol. Biol.*, **138**, 321–333.

Cilibrasi,R., de Wolf,R. and Vitanyi,P. Algorithmic clustering of music. (in press).

Goldman,D., Istrail,S. and Papadimitriou,C. (1999) Algorithmic aspects of protein structure similarity. *Proceedings of the 40th Annual Symposium on Foundations of Computer Sciences*, IEEE, Computer Society, pp. 512–522.

Goldman,D. (2000) Algorithmic Aspects of Protein Folding and Protein Structure Similarity, PhD Thesis, Department of Computer Sciences, UC Berkeley.

Gordon,A.D. (1999) *Classification*, 2nd edn. Chapman and Hall/CRC.

Grumbach,S. and Tahi,F. (1994) A new challenge for compression algorithms: genetic sequences. *J. Info. Process. Manag.*, **30**, 866–875.

Holm,L. and Sander,C. (1993) Protein-structure comparison by alignment of distance matrices. *J. Mol. Biol.*, **233**, 123–138.

Holm,L. and Sander,C. (1996) Mapping the protein universe. *Science*, **273**, 595–602.

Hubbard,T.J.P. (1999) Rms/coverage graphs: a qualitative method for comparing three-dimensional protein structure predictions. *Prot. Struct. Funct. Genet.* **3** (suppl.), 15–21.

Koehl,P. (2001). Protein structure similarities. *Curr. Opin. Struct. Biol.*, **11**, 348–353.

Krasnogor,N. (2002) In *Studies on the Theory and Design Space of Memetic Algorithms*. PhD Thesis, University of the West of England, Bristol, UK.

Krasnogor,N. (2003) Self-generating metaheuristics in bioinformatics: The proteins structure comparison case. *J. Genet. Program. Evolv. Mach.*, **5**.

Li,M., Badger,J.H., Chen,X., Kwon,S., Kearney,P. and Zhang,H. (2001) An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, **17**, 149–154.

Li,M., Chen,X., Li,X., Ma,B. and Vitanyi,P. (2003) The similarity metric. *Proceedings of the 14th ACM-SIAM Symposium Discrete Algorithms* (*SODA*) *2003*, SIAM/ACM Press.

Lancia,G., Carr,R., Walenz,B. and Istrail,S. (2001) 101 optimal pdb structure alignments: a branch-and-cut algorithm for the maximum contact map overlap problem. *Proceedings of The Fifth Annual International Conference on Computational Molecular Biology, RECOMB 2001*.

Lackner,P., Koppensteiner,A., Domingues,F.S. and Sippl,M.J. (1999) Automated large scale evaluation of protein structure predictions. *Prot.: Struct. Funct. Genet.*, **3** (suppl.), 7–14.

Leluk,J., Konieczny,L. and Roterman,I. (2003) Search for structural similarity in proteins. *Bioinformatics*, **19**, 117–124.

Li,M. and Vitanyi,P. (1997) *An Introduction to Kolmogorov Complexity and Its Applications*. Springer.

May,A.C.W. (1999) Towards more meaningful hierarchical classification of aminoacids scoring matrices. *Prot. Struct. Funct. Genet.*, **37**, 20–29.

Murzin,A.G., Brenner,S.E., Hubbard,T. and Chothia,C. (1995) Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Maiorov,V.N. and Crippen,G.M. (1994) Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins. *J. Mol. Biol.*, **235**, 625–634.

Orengo,C.A., Michie,A.D., Jones,S., Jones,D.T., Swindells,M.B. and Thornton,J.M. (1997) Cath—a hierarchic classification of protein domain structures. *Structures*, **5**, 1093–1108.

Siew,N., Elofsson,A., Rychlewsky,L. and Fischer,D. (2000) Maxsub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, **16**, 776–785.

Taylor,W.R. (1999) Protein structure comparison using iterated double dynamic programming. *Prot. Sci.*, **8**, 654–665.

Varre,J.S., Delahaye,J.P. and Rivals,E. (1998) The transformation distance: a dissimilarity measure based on movements of segments. *German Conference on Bioinformatics*. Koel, Germany.

Wu,T.D., Schmidler,S.C., Hastie,T. and Brutlag,D.L. (1998) Regression analysis of multiple protein structures. *J. Comput. Biol.*, **5**, 585–595.

Zemla,A. (2000) Lga program: a method for finding 3-d similarities in protein structures. *Nucleic Acids Res.*, **31**, 3370–3374.

Zemla,A., Venclovas,C., Moult,J. and Fidelis,K. (1999) Processing and analysis of casp3 protein structure predictions. *Prot. Struct. Funct. Genet.*, **3** (suppl.), 22–29.