Mathware & Soft Computing 16 (2009) 87-102

Fuzzy Contact Maps Overlap for Protein Comparison: the Roles of Fitness and Normalization and the relation with Crisp Contact Maps

Juan R. González, David A. Pelta, Lluvia Morales
Department of Computer Science and Artificial Intelligence,
University of Granada, 18071 Granada, Spain
e-mail: {jrgonzalez,dpelta}@decsai.ugr.es,
lluviamorales@ugr.es

Abstract

The comparison of protein structures is an important problem in Bioinformatics, and Soft Computing techniques were recently introduced for achieving a better representation and, potentially, for getting better solving strategies. In this paper we propose new alternatives for measuring the cost in the Generalized Maximum Fuzzy Contact Map Overlap problem, and we analyze the role of normalization when protein classification is performed. Moreover, we will also compare the optimization performance of the direct application of crisp contact maps over solving the problems through the fuzzy contact maps model. This extends our previous works where it was preliminarily shown that the resolution of the fuzzy model can sometimes lead to better crisp values than the ones obtained solving the crisp model directly.

1 INTRODUCTION

A protein is a complex molecule composed by a linear arrangement of amino acids. Each amino acid is a multi-atom compound. Usually, only the "residue" part of these amino acids are considered when studying protein structures for comparison purposes. Thus a protein's *primary sequence* is usually thought-of as composed of "residues". Under specific physiological conditions, the linear arrangement of residues will *fold* and adopt a complex three dimensional shape. The shape thus adopted is called the *native state* (or tertiary structure) of the protein. In its native state, residues that are far away along the linear arrangement may come into proximity in three dimensional space in a fashion similar to what occurs with the extremes of a sheet of paper when used to produce complex origami shapes. The proximity relation between residues in a protein can be captured by a mathematical construct called a "contact map".

A contact map [9, 11] is a concise representation of a protein's 3D structure. Formally, a map is specified by a 0-1 matrix S, with entries indexed by pairs of protein residues, such that:

$$S_{i,j} = \begin{cases} 1 & \text{if residue } i \text{ and } j \text{ are in contact} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Residues i and j are said to be in "contact" if their Euclidean distance is at most $\Re$ (a threshold measured in Angstroms) in the protein's native fold.

The comparison of proteins through their contact maps is equivalent to solving the maximum contact map overlap problem (MAX-CMO)[5, 4, 14], a problem that belongs to the NP-Hard class of complexity.

Since the amount of protein data available is increasing at a fast rate (with about 40000 structures currently present on the Worldwide Protein Data Bank [3]), the problem of comparing a new protein against all the known proteins or even with a subset or a specific representative database is a very big task. So, despite there exist exact algorithms for the MAX-CMO model [14] they can not be applied to most instances because the amount of resources required would be prohibitive.

These considerations led us to a first use of Soft Computing techniques to contribute in the field by developing a simple and fast heuristic that could obtain good results for this model and provide biologically relevant solutions without the need to find the exact solutions to the model. This heuristic has been published recently on [7] where it is extensively tested showing how the proposed algorithm, that is based on the Variable Neighborhood Search metaheuristic, can both obtain near-optimal results and solutions and similarity values that are biologically relevant for the purpose of classification. The heuristic is also shown to be competitive in classification performance with similarity measures coming from methods that compare proteins through different models like the ones based on distance matrixes [10, 1, 7].

But although crisp contact maps are useful to compare proteins, it is also known that the errors on the determination of the 3D coordinates of the residues of a protein by X-Ray crystallography or NMR range from 0.01 to 1.27Å [8], which is close to the value of some covalent bonds. This kind of imprecision can not be modeled through the crisp contact maps but there exists an alternative formulation that uses fuzzy contact maps and a new model to compare such maps: the generalized fuzzy contact map overlap problem GMAX-FCMO [12], thus making another technique from Soft Computing (fuzzy sets) come into play. The use of fuzzy contact maps allows to soften the thresholds for the contacts to take into account the potential errors in the determination of coordinates and it also serves as a way to give different semantics to contacts that arise at different distance ranges.

But despite of its high potential GMAX-FCMO still needs much more research to be in pair with the research done in MAX-CMO, so in this paper we are going to extend our previous works on it [6]. Firstly, by exploring the application of the GMAX-FCMO model in protein structure classification considering two aspects: a) the use of new alternatives to measure the cost of a solution, and b) the role of the normalization of the fitness. Secondly, in the previous works we also addressed

the comparison of fuzzy contact maps against crisp contact maps and we showed that if we first solved the problem through GMAX-FCMO model, and then such solutions were measured as in MAX-CMO, the results obtained could be better than those obtained when MAX-CMO is solved directly. Here we will also extend that analysis with a different dataset.

The paper is organized as follows: In section 2, fuzzy contact maps as well as their comparison on the MAX-CMO and GMAX-FCMO models is presented. Section 3 describes the experiments and results obtained. Finally, section 4 is devoted to the conclusions and future work.

2 FUZZY CONTACT MAPS MODEL DESCRIPTION

Fuzzy contact maps were introduced in [12] with two aims: a) to take into account potential measurement errors in atom coordinates, and b) to allow highlighting features that occur at different thresholds.

We define a *fuzzy contact* as a contact made by two residues that are *approximately*, rather than exactly, at a distance $\Re$. Formally, a fuzzy contact is defined by:

$$F_{i,j} = \mu(\overline{[i,j]}, \Re) \tag{2}$$

where $\mu()$ is a particular definition of (fuzzy) contact, $\overline{[i,j]}$ stands for the Euclidean distance between residues i, j, and $\Re$ is the threshold as for the crisp contacts. The standard, i.e. crisp, contact map is just a special case of the fuzzy contact map when a user-defined α-cut is specified.

Figure 1 (a), (b) and (c) shows three alternative definitions for "contact". Each panel in the Figure is a fuzzy contact map where a dot appears for each pair of residues having $F_{i,j} > 0$ (i.e. the support of the corresponding fuzzy set).

Fuzzy contact maps are further generalized by removing the constraint (in the original model) of having only one threshold $\Re$ as a reference distance. In this way, besides having a membership value, a contact will have a "type". The formal definition of a General Fuzzy Contact is given by:

$$F_{i,j} = max\{\mu_1(\overline{[i,j]}, \Re_1), \ldots, \mu_n(\overline{[i,j]}, \Re_n)\} \tag{3}$$

with the contact map C defined as:

$$C^{r \times r} = (F_{i,j}) \; with \; 0 \leq i, j \leq r \tag{4}$$

i.e. up to n different thresholds and up to n different semantic interpretations of "contact" are used to define the $r \times r$ contact map being r the number of residues in the protein.

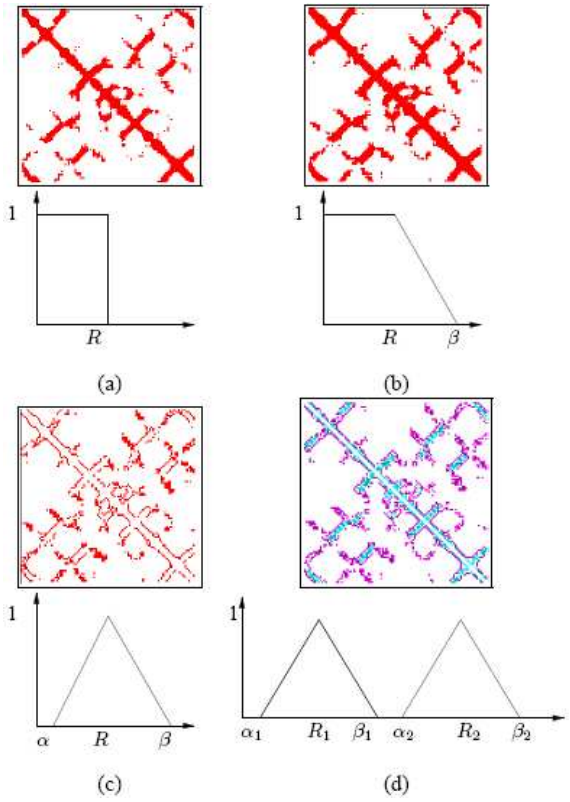


Figure 1: Four examples of contact maps. In (a) the standard model; (b) the simplest fuzzy generalization; (c) another generalization; (d) a two threshold, two membership functions fuzzy contact map.

2.1 PROTEIN COMPARISON THROUGH CONTACT MAPS OVERLAP

A solution for the comparison of two contact maps under the crisp Maximum Contact Map Overlap model consists of an alignment or pairing between the nodes of the two contact maps. The value of an alignment is the number of cycles of length four that appear between the corresponding graphs after the pairings. This value is called the overlap (fitness) of the two contact maps and the goal is to maximize it.

For example, Figure 2 shows a sample solution for the comparison of two contact maps of 5 and 7 residues respectively. In the crisp model, we can omit the colors of the arcs. Three residues are paired, as shown with a dotted line (first index corresponds to the bottom graph): $1 \leftrightarrow 1$, $2 \leftrightarrow 4$ and $3 \leftrightarrow 5$, and the overlap value is two because there are two cycles of length four.

The first cycle is formed by the pairing $1 \leftrightarrow 1$, the arc $1 \leftrightarrow 5 \in P_2$, the pairing $3 \leftrightarrow 5$ and the arc $3 \leftrightarrow 1 \in P_1$. The second one begins with the pairing $2 \leftrightarrow 4$, follows the arc $4 \leftrightarrow 5 \in P_2$, then the pairing $3 \leftrightarrow 5$ and finally, the arc $3 \leftrightarrow 2 \in P_1$.

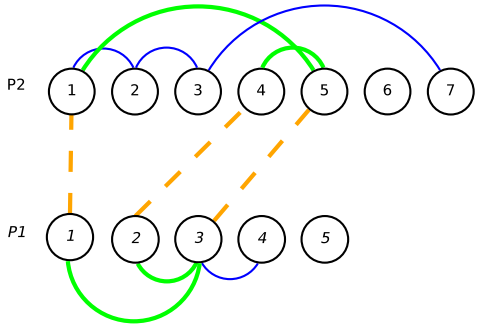


Figure 2: Two levels of contacts in a fuzzy contacts graph.

Solutions for the GMAX-FCMO model have exactly the same structure as in the crisp MAX-CMO, but the overlap value is computed differently. Now, the arcs have weights and types (colors), so the contribution of each cycle to the global fitness is calculated as a function of the membership values of the contacts involved and their types. In the original GMAX-FCMO model, both membership values are multiplied and then, if both contacts have the same type, then the contribution is added to the fitness. Otherwise, when contacts of different type are involved in a cycle, the contribution is subtracted from the fitness. In this way, alignments between contacts of different types are penalized.

So, as the contact maps in Figure 2 are fuzzy, the contribution of one cycle to the fitness is the product of the membership value of the contact $1 \leftrightarrow 3$ of the bottom protein and that of the contact $1 \leftrightarrow 5$ of the upper protein, with positive sign because the two contacts are of the same type; and this value is added to the product of the membership of contact $2 \leftrightarrow 3$ and the membership of contact $4 \leftrightarrow 5$.

3 EXPERIMENTS AND RESULTS

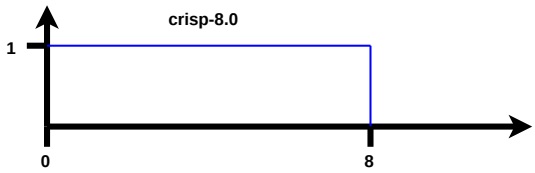
Two experiments will be presented. The first one explores two factors and their influence on the protein structure classification performance: 1) the use of different alternatives of measuring the cost of a solution, and 2) the role of normalization. The second experiment will be devoted to the comparison of the optimization of the crisp MAX-CMO values when the MAX-CMO model is solved directly against the results that can be obtained if we solve first the GMAX-FCMO model and then convert the value of its solutions to the crisp model values.

All the computational experiments will follow the same scheme: we will conduct queries on a protein database, solving a GMAX-FCMO problem for comparing the query with each protein in the database. Then, we will have a list of solutions

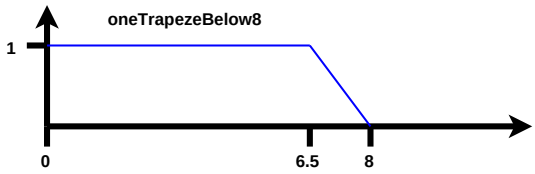
with their respective overlap values that can further be normalized or converted to the crisp equivalent value. The protein database used is a selection of the protein structures that belong to the Nh3D v3.0 test dataset [13]. The Nh3D dataset has been compiled by selecting well resolved representatives from the Topology level of CATH database and contains 806 topology representatives belonging to 40 architectures, that can be further classified in terms of "class". Since there is only one protein for each topology and one of the main interests of this paper is to assess classification performance we need to focus on the higher CATH levels (architecture and then class). This was the reason to select only a subset from the Nh3D database as we wanted to have an evenly distributed database where we had several representative proteins for each architecture and not some architectures with a lot of proteins and some with just one or few. This serves to ensure that the results apply to a broader range of different proteins since each architecture will now have a similar weight (effect) in the results. Therefore, fist we excluded all the architectures that dit not have at least 10 topology representatives on it so only 15 architectures remained. Then we selected as the query set the structures that have the nereast to average size for each of these 15 architectures. Finally, the test database was made by taking all the proteins of this 15 architectures, removing the query proteins and picking randomly 10 proteins of each architecture leading to a total of 150 proteins on the test database.

The resolution of the GMAX-FCMO between the queries and every protein in the test database was done with an adapted version of a Multi-Start Variable Neighborhood Search (VNS) metaheuristic developed for MAX-CMO that was presented in [7]. It is also publicly available online as one of the methods used in the ProCKSi server [2]. This algorithm follows a standard VNS algorithm structure with just a few changes: there is an extra Multi-Start loop to better explore the solution space and reduce the big influence of a single initial solution; and a "simplify" function that is used after every local search to remove pairings that do not contribute to the solution fitness (it helps to avoid the saturation of the solution with useless pairings). The algorithm also uses reduced solution evaluation to recompute the cost of a neighbor solution considering only the changes from the current one, thus significantly reducing the computational time needed.

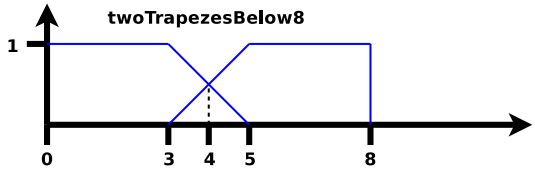
But since we are comparing proteins through contact maps the proteins in the database need to be converted to contact maps before the comparison can be done. To construct the contact maps we will consider three different membership functions for the contacts as shown on Figure 3, all of which cover the same distance range (in Å) so the contact maps of a given protein for the three definitions will be the same except for the specific membership levels and types of the contacts. The first function (a) is equivalent to a standard MAX-CMO type of contact map with a 8Å threshold. The second one (b) is the simplest fuzzy generalization that has a decaying slope that reduces the level of membership of a contact as the distance approaches from 6.5 to 8Å. The third function (c) is a fuzzy function that distinguish between two different types of contacts for residues at short or long distances.



(a) Crisp function



(b) Fuzzy function with one type of contact



(c) Fuzzy function with two types of contacts

Figure 3: Experimental functions.

3.1 EXPERIMENT 1: ASSESSING THE EFFECT OF DIFFERENT GMAX-FCMO COST CALCULATIONS AND NORMALIZATIONS IN PROTEIN STRUCTURE CLASSIFICATION

This first experiment will analyze the effect of different alternatives for the calculation of the cost of a solution in the GMAX-FCMO model and the application of a normalization to the resulting overlap values.

3.1.1 ALTERNATIVES FOR COST CALCULATIONS

The value of an overlap is the sum of the contribution of every cycle of length four. In the crisp model, every cycle contributes a unity to the overlap.

Cycles in the fuzzy model has the appearance shown in Fig. 4. The contribution of a cycle is calculated as $C = \mu(a,b) \times \mu(c,d) \times F(t(a,b),t(c,d))$, where $t(a,b), t(c,d)$ stand for the type (color) of the contact between a,b and c,d respectively. The function F simply returns 1, if both contacts are of the same type, and -1 in other case. So, the cost of the cycles in the example are $0.8 \times 0.5 \times 1$ in Fig. 4 (left) and $0.8 \times 0.5 \times -1$ in the right cycle. Here we are going to consider four different alternatives to measure the contribution of an individual cycle, namely:

1. Product: $C = \mu(a,b) \times \mu(c,d)) \times F(t(a,b),t(c,d))$

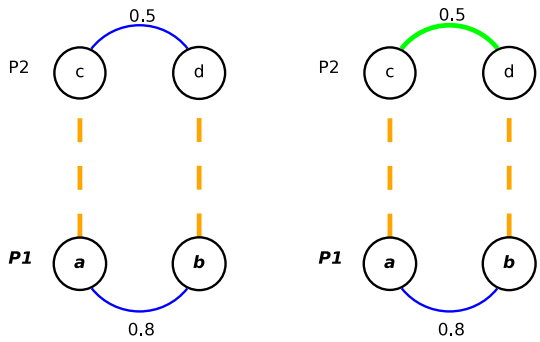


Figure 4: A cycle of length four making contact of the same (left) and different (right) types.

2. Min: $C = min(\mu(a,b), \mu(c,d)) \times F(t(a,b), t(c,d))$
3. Max: $C = max(\mu(a,b), \mu(c,d)) \times F(t(a,b), t(c,d))$
4. Avg: $C = ((\mu(a,b) + \mu(c,d))/2) \times F(t(a,b), t(c,d))$

where we still have the F function that provides a negative or positive sign for the cycle contribution depending if the contacts are of the same type or not (semantic mismatch). But the actual absolute cycle contribution value is computed now in four different ways. For comparison purposes the first alternative remains equal to the original GMAX-FCMO cycle contribution.

3.1.2 NORMALIZATION ALTERNATIVES

Overlap values per se, are not allways useful (at least in the crisp model) for classification purposes, as such values depend on the size of the proteins being compared. Once the GMAX-FCMO is solved, a normalization scheme should be applied and it is claimed that this scheme may play a crucial role in protein classification.

Following the ideas posed in [6, 7], we will use four alternatives in our experiments:

1. $Norm1(P_i, P_j) = overlap(P_i, P_j) \ / \ min(contactsP_i, contactsP_j)$
2. $Norm2(P_i, P_j) = 2 \ * \ overlap(P_i, P_j)/(contactsP_i \ + \ contactsP_j)$
3. $Norm3(P_i, P_j) = \begin{cases} 0 & \text{if contacts difference is greater than } 75\% \\ Norm1(P_i, P_j) & \text{otherwise} \end{cases}$
4. $NormFuzzy(P_i, P_j) = overlap(P_i, P_j) \ / \ max(selfSim(P_i), \ selfSim(P_j))$

where the self-similarity ($selfSim$) of a protein is the value of the optimal overlap of a protein with itself. The computation of this self-similarity is trivial since it simply coincides with the number of contacts in the protein for the crisp case. For computing the self-similarity on the fuzzy case just the single alignment that pairs every residue of the protein with itself needs to be considered.

3.1.3 COMPUTATIONAL EXPERIMENTS

In this experiment we perform a comparison of the queries of our database (as described at the beginning of 3) against the whole test database for every cost calculation alternative and for two definitions of contact maps (oneTrapezeBelow8 and twoTrapezesBelow8) as given in Figure 3 (b) and (c).

The computational experiments are conducted by running our VNS metaheuristic over every different contact map pair leading to a list of overlap values that will be normalized with every normalization alternative proposed. To evaluate the classification performance the results are analyzed using ROC curve analysis and the area under the curve (AUC) values, both in terms of classification at the level of architecture and at the level of class. To be able to apply a ROC analysis to the protein classification several simple steps are taken. First of all, each combination of cost calculation, normalization scheme and fuzzy function definition is considered to be a different classifier to which the ROC analysis needs to be applied, where the similarity (the "normalized" overlap obtained for each protein pair) is considered as the classifier value for that pair. Then, since the classification will be done at the CATH class and architecture levels and both of them have more than one possible value (classes in a general classifier sense, not to be confused with the CATH classes) we need to construct and additional binary class for the ROC analysis. This binary class will be used as the state variable for the ROC analysis and it simply indicates if the two proteins on any pairwise comparison are of the same or different type (classifier class) at the CATH similarity level being considered (class or architecture). In this manner, it is possible to perform a ROC analysis that assesses the performance of the classifiers in terms of their capacity to provide higher similarity values for proteins of the same type.

Tables 1 and 2 show the AUCs for every cost calculation and normalization alternative when using contact maps according to the oneTrapezeBelow8 definition. The respective AUCs when the contact maps come from the twoTrapezesBelow8 definition are shown on Tables 3 and 4.

Table 1: AUCs for the classification at the level of arquitecture (contact map definition = oneTrapezeBelow8).

	Fitness	**Norm1**	**Norm2**	**Norm3**	**NormFuzzy**
PRODUCT	0.565	0.468	0.625	0.542	0.622
MIN	0.571	0.479	0.637	0.552	0.631
MAX	0.565	0.470	0.628	0.546	0.623
AVG	0.569	0.476	0.636	0.551	0.629

As we can see by looking at any column of the tables, the AUC values do not change significantly for any of the proposed cycle contributions. More precisely, the AUC values within each column have differences of at most 0.011, so the classification performance is mostly unaffected whatever the cycle constribution is. Considering these results it is clear that the actual value for the contribution of a cycle is not important at all. This probably comes from the fact that the

Table 2: AUCs for the classification at the level of class (contact map definition = oneTrapezeBelow8).

	Fitness	**Norm1**	**Norm2**	**Norm3**	**NormFuzzy**
PRODUCT	0.569	0.419	0.553	0.491	0.580
MIN	0.573	0.426	0.557	0.498	0.584
MAX	0.574	0.420	0.561	0.494	0.585
AVG	0.575	0.426	0.561	0.497	0.586

Table 3: AUCs for the classification at the level of arquitecture (contact map definition = twoTrapezesBelow8).

	Fitness	**Norm1**	**Norm2**	**Norm3**	**NormFuzzy**
PRODUCT	0.565	0.471	0.630	0.547	0.625
MIN	0.571	0.482	0.643	0.557	0.630
MAX	0.567	0.475	0.635	0.551	0.627
AVG	0.574	0.490	0.646	0.565	0.636

VNS algorithm will try to pair any residues that lead to more contributing cycles. Therefore, all the relevant pairings will get added as all of them improve the solution. In this manner, the relative ordering of the values of solutions obtained with the VNS will remain almost unchanged with all the different cycle contributions as the effect will be similar for all solutions (pairs). This is the reason why the classification performance (AUC values) is almost identical. For this experiment there is also no difference in classification performance independently of which of the two definitions of contact maps is being used (Tables 1-2 versus Tables 3-4). The differences between the two definitions are only in the order of thousandths so we can say that they both lead to an equally good ranking of similarity among proteins.

To continue the analysis of the tables and having determined that the cost calculation and the contact map definition do not affect the results in any relevant way we now focus on the effect of normalization. It is easily seen that as we move through the different normalizations on any of the tables (the different values on any row) the differences are quite noticeable, with *Norm2* and *NormFuzzy* normalizations as the best options for post processing the overlap in order to better classificate the proteins in the database.

This statement can be seen visually in Figure 5 for proteins with the same architecture and in Figure 6 for proteins with the same class. These results are similar to the results obtained for the crisp model [7] in the sense that the normalization done is again an important factor on the fuzzy model. The differences are very significative among the normalizations with the aforementioned *Norm2* and *NormFuzzy* having reasonably good classification performance at the same time that *Norm1* and *Norm3* lead to worse than random performance (AUCs below

Table 4: AUCs for the classification at the level of class (contact map definition = twoTrapezesBelow8).

	Fitness	**Norm1**	**Norm2**	**Norm3**	**NormFuzzy**
PRODUCT	0.572	0.417	0.561	0.491	0.576
MIN	0.576	0.424	0.566	0.498	0.580
MAX	0.578	0.418	0.570	0.499	0.583
AVG	0.575	0.424	0.564	0.496	0.579

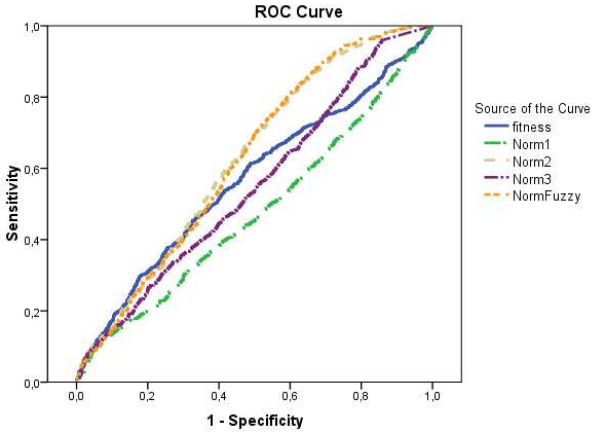


Figure 5: ROC curves for the same architecture of proteins (contact map definition = oneTrapezeBelow8, cost calculation = AVG).

0.5).

3.2 EXPERIMENT 2: ON THE RELATION BETWEEN FUZZY AND CRISP OVERLAP VALUES

In experiment one we have shown how the normalization of the overlap values was a very important factor in the protein structure classification of Max-CMO while the different alternatives of cost calculation and the fuzzy contact map definition used had no effect.

In this second experiment we want to compare the use of different contact map definitions not for their effects on classification but in terms of the optimization values that can be obtained from them. In previous works [6] we have already done a comparison of the effect of the same three fuzzy functions definitions presented on this paper but on a very small dataset. Now we are going to extend that analysis on the bigger database considered here.

First of all we should recall that from the definitions for MAX-CMO and GMAX-FCMO, it is easy to infere that the fitness values obtained for the crisp-

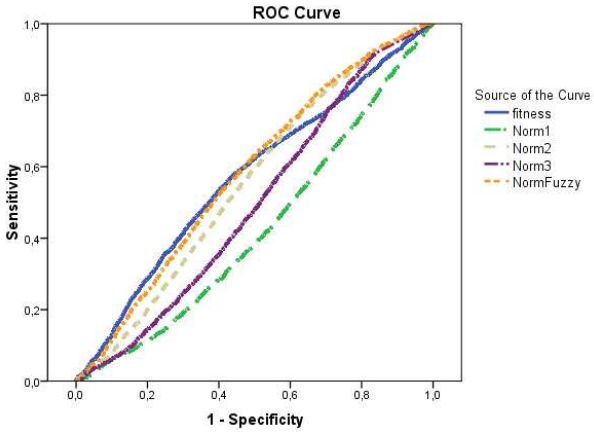


Figure 6: ROC curves for the same class of proteins (contact map definition = oneTrapezeBelow8, cost calculation = AVG).

8.0 function are an upper bound for the fitness values that oneTrapezeBelow8 and twoTrapezesBelow8 fuzzy functions can achieve (independently of which of the presented alternatives of cost calculation is chosen). In the former, every cycle of length four, increases the overlap by 1, while in the fuzzy case, the contribution of a cycle may vary in the range [-1, 1]. Due the different scales of the overlap values, the comparison among values from different map definitions is not easy. However, we can compare them using a simple procedure [6]: given a particular solution (an alignment) obtained when solving a GMAX-FCMO instance, we can calculate the equivalent crisp fitness for the best solution found, so the final alignment solution obtained with either oneTrapezeBelow8 or twoTrapezesBelow8 can then be compared against the ones obtained when crisp-8.0 function is used. That is to say: when we solve the GMAX-FCMO we obtain an alignment between the graphs of the two contact maps under the fuzzy definition. If we then ignore the types of the contacts and the membership values, we have a valid alignment for the contact maps coming from the crisp-8.0 definition whose cost can then be computed and compared to the one obtained when solving crisp-8.0 directly.

In the previous work we found that the optimization values obtained with crisp-8.0 were statistically better than those obtained with oneTrapezeBelow8. However, the comparison of crisp-8.0 against twoTrapezesBelow8 and oneTrapezeBelow8 against twoTrapezesBelow8 lead to no statistical difference. Finally we concluded there that the crisp values obtained through solving GMAX-FCMO can sometimes be better than those obtained when solving MAX-CMO directly and we showed that the difference of size between the contact maps compared could be used as one of the factors to determine if one resolution method or the other should be chosen.

To see if that previous results also hold with the new database we had also performed an ANOVA analysis to see if there were significant differences between the

resolution with the three types of contact map definitions. This time the ANOVA test showed no statistically difference at all between any of them. Therefore we can conclude that again with this dataset the resolution of MAX-CMO through GMAX-FCMO is possible. That is to say that the crisp values obtained when converting fuzzy values to crisp are of the same quality that the crisp values obtained solving MAX-CMO directly even when the optimization goal is quite different. It is clear then that the GMAX-FCMO model of introducing semantic for contact has sense because when we pair only similar contacts the level of quality of the solutions obtained is not only good in GMAX-FCMO sense but it also remains good when it is considered in crisp (MAX-CMO) sense.

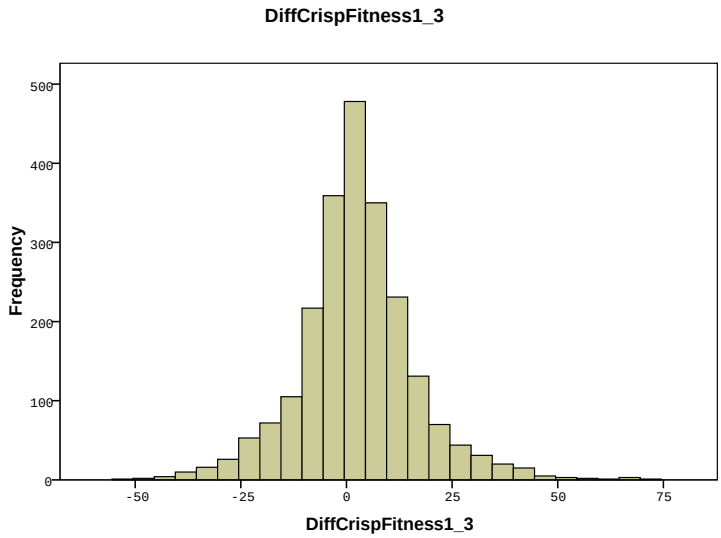


Figure 7: Histogram of frequencies of the cost differences between crisp-8.0 and twoTrapezesBelow8.

To further see how the crisp value results from the different definitions compare, Figure 7 shows a histogram of frequencies of the cost differences between crisp-8.0 and twoTrapezesBelow8. It is clear from the histogram that both definitions lead to values that outperform the other in a comparable amount of times. Moreover, despite the fact that the differences are usually small they can also sometimes be quite big in favour of either side. Similar histograms are obtained on the comparison of crisp-8.0 against oneTrapezeBelow8 and oneTrapezeBelow8 against twoTrapezesBelow8 so it is clear that no definition is the best one for every protein structure comparison instance.

Figure 8 shows a scatter plot of the differences of crisp cost obtained by the crisp-8.0 and twoTrapezesBelow8 functions against the number of contacts in the

contacts maps being compared and with the results separated by the class of the query proteins (classP1). A point in the graph is positive (displayed as a circle) when the crisp fitness of crisp-8.0 was better, and negative (displayed as a rhombus) when the crisp fitness coming from the conversion of the fuzzy solutions obtained with the twoTrapezesBelow8 function was the better one.

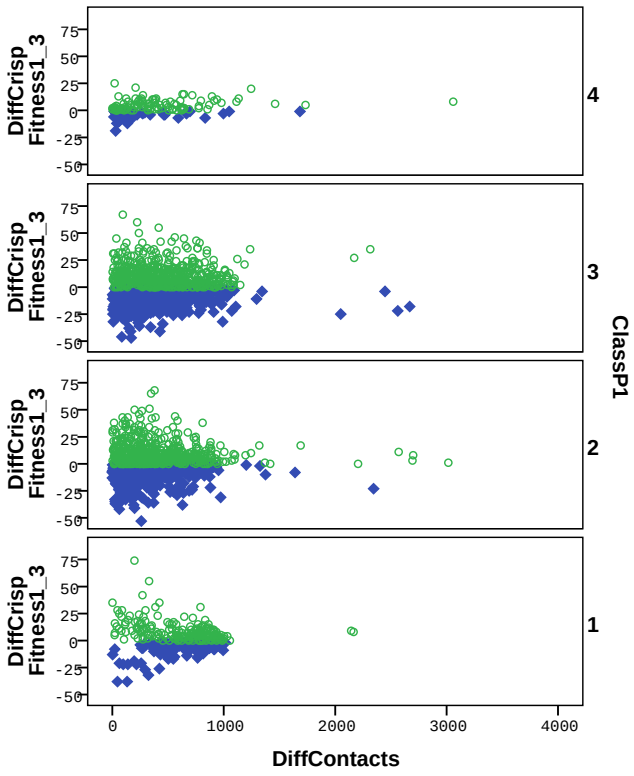


Figure 8: Dispersion of cost differences between crisp-8.0 and twoTrapezesBelow8 against difference of contacts.

The graph shows again that twoTrapezesBelow8 and crisp-8.0 outperform each other a similar number of times (even when the value of crisp-8.0 should be an upper bound of the twoTrapezesBelow8 value). This is possible because the problems are solved heuristically due to the high cost of the exact methods. Therefore using a different fuzzy function will lead to contact maps that may be easier to solve with the VNS. Also, unlike what occurred in previous works the number of contacts does not seem to affect since the number of times that twoTrapezesBelow8 outperforms crisp-8.0 is almost homogeneously distributed independently of the size difference of the proteins compared, so the findings about this on [6] need to be reconsidered and

tested more throughly with more datasets. Moreover, new ways to determine when to apply MAX-CMO and when to apply GMAX-FCMO should be investigated.

4 CONCLUSIONS

This paper has analyzed the influence of the computation of the contribution of each cycle and the normalization scheme when doing protein classification using the GMAX-FCMO model for protein structure comparison.

The results indicate that the strategy used to compute the contribution of each cycle to the solution is not relevant while the normalization is playing a key role. This emphasizes the importance of normalization as it has been proved to be very important both for the crisp and the fuzzy model.

We have also compared the resolution of the protein structure comparison with MAX-CMO against the resolution with GMAX-FCMO in terms of the crisp (MAX-CMO) costs obtained. We have seen how the previous results about the resolution through GMAX-FCMO being able to outperform the direct resolution with MAX-CMO also hold for this new dataset. There is no statistical difference between them and there are many cases where any of them outperforms the other. Nevertheless, the previous findings that indicated that the number of contacts may possibly be used to predict when to use one method over the other does not hold for the experiments performed here. Therefore, we plan to extend our research to try to determine if it is possible or not to use some characteristics of the proteins for selecting one model over the other as the optimization technique on any specific protein structure comparison instance.

Our future work plans also include to conduct more analysis on the fuzzy model to try to find some good guidelines on how to define each type of contacts both regarding the number of types to use and the specific distance values (in Angstroms) to associate to them.

Acknowledgements

This work is supported in part by Project HeuriCosc (TIN2005-08404-C04-01) from the Spanish Ministry of Education and Science; and Projects MINAS (TIC-00129-JA) and P07-TIC02970 from the Regional Government of Andalusia.

L. Morales is supported by the Mexican National Council on Science and Technology (CONACYT).

References

[1] Z. Aung and K.-L. Tan. Matalign: Precise protein structure comparison by matrix alignment. *Journal of Bioinformatics and Computational Biology*, 4(6):1197–1216, 2006.

[2] D. Barthel, J. D. Hirst, J. Blazewicz, E. K. Burke, and N. Krasnogor. ProCKSI: a decision support system for Protein (Structure) Comparison, Knowledge, Similarity and Information. *BMC Bioinformatics*, 8:416, 2007.

[3] H. Berman, K. Henrick, H. Nakamura, and J. L. Markley. The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. *Nucl. Acids Res.*, 35(suppl_1):D301–303, 2007.

[4] A. Caprara, R. Carr, S. Istrail, G. Lancia, and B. Walenz. 1001 optimal pdb structure alignments: Integer programming methods for finding the maximum contact map overlap. *Journal of Computational Biology*, 11(1):27–52, 2004.

[5] R. Carr, W. Hart, N. Krasnogor, J. Hirst, E. Burke, and J. Smith. Alignment Of Protein Structures With A Memetic Evolutionary Algorithm. In *GECCO-2002 Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1027–1034. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2002.

[6] J. R. González and D. A. Pelta. On using fuzzy contact maps for protein structure comparison. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). DOI: 10.1109/FUZZY.2007.4295614*, 2007.

[7] J. R. González, D. A. Pelta, and M. Moreno-Vega. A simple and fast heuristic for protein structure comparison. *BMC Bioinformatics*, 9:161, 2008.

[8] R. A. Laskowski. Structural quality assurance. In P. Bourne and H. Weissig, editors, *Structural Bioinformatics*. Wiley-Liss, Inc, 2003.

[9] S. Lifson and C. Sander. Antiparallel and parallel bold italic beta-strands differ in amino acid residue preferences. *Nature*, 282:109–111, 1979.

[10] H. Liisa and J. Park. DaliLite workbench for protein structure comparison. *Bioinformatics*, 16(6):566–567, 2000.

[11] L. Mirny and E. Domany. Protein fold recognition and dynamics in the space of contact maps. *Proteins Structure Function and Genetics*, 26(4):391–410, 1996.

[12] D. Pelta, N. Krasnogor, C. Bousono-Calzon, J. L. Verdegay, J. Hirst, and E. Burke. A fuzzy sets based generalization of contact maps for the overlap of protein structures. *Journal of Fuzzy Sets and Systems*, 152(1):103–123, 2005.

[13] B. Thiruv, G. Quon, S. A. Saldanha, and B. Steipe. Nh3d: a reference dataset of non-homologous protein structures. *BMC Struct Biol*, 5, 2005.

[14] W. Xie and N. V. Sahinidis. A reduction-based exact algorithm for the contact map overlap problem. *Journal of Computational Biology*, 14(5):637–654, 2007.