IL NUOVO CIMENTO Vol. 35 C, N. 5 Suppl. 1 Settembre-Ottobre 2012
DOI 10.1393/ncc/i2012-11332-4

Colloquia: PR PS BB 2011

Protein structure analysis through Hough Transform and Range Tree

V. Cantoni([1]) and E. Mattia([2])

([1]) *Department of Computer Engineering and Systems Science*
University of Pavia - Pavia, Italy
([2]) *Center for Systems Chemistry, Rijksuniversiteit Groningen - Groningen, The Netherlands*

ricevuto il 30 Settembre 2011; approvato l' 1 Dicembre 2011

Summary. — The Generalized Hough Transform (GHT) allows to recognize general patterns once defined a model to be recognized, a reference point (RP) rigid with the model, and a mapping rule. This rule establishes the contributions in the parameters space; this space, generally speaking, is given by the parameters of a rigid motion leading to overlap a model item with an equal item detected on the unknown pattern. In this paper we introduce the GHT applied to motifs, domains and entire proteins retrieval into a protein data base. The spatial attitude of a single protein secondary structure (SS) constitutes the item supporting the contributions. If the unknown pattern contains a block of N SS of the model to be recognized, the N corresponding votes will have a common point, so accumulating N contributions. An analysis of the neighborhoods around the areas with high contributions density is necessary. It is not sufficient and often inaccurate to limit the analysis to the peaks even if the number of contribution is closed to the expected one. Both convenient data structures for effectively operating in the neighborhoods (a range tree data structure) and suitable decision criteria have been introduced. Preliminary results of comparative analysis are given.

PACS 87.18.Xr – Proteomics.
PACS 87.85.mk – Proteomics.
PACS 87.15.B- – Structure of biomolecules.
PACS 87.15.bd – Secondary structure.

1. – Definition

The Hough Transform can be used efficiently in the context of protein structural comparison and motif retrieval, thereby constituting another fundamental tool among the available heuristics that allow to estimate the presence of particular structural features within the structure of a protein.

2. – Introduction

Systems Biology benefits enormously from automatic methods aimed at sorting out meaningful information and overall trends from the huge amount of data about biological systems that has been, is being and will be collected with modern high-throughput techniques. Among these methods, it is widely acknowledged that a substantial role in carrying out data mining nowadays is played by protein structural comparison. Known functional units such as structural motifs can be retrieved in recently discovered proteins and similarities between known and new structures can be established. This allows inferring protein function, explaining the role of specific sequences in biochemical pathways and biological networks, building up phylogenetic trees and ultimately creating new database annotations, which in including the results of the completed predictions will be helpful for the structures that will be discovered in the future. Various approaches have been developed for protein structure comparison, based on either distance matrices [1, 2] graph theory or geometric hashing [3, 4]. The various available algorithms are heuristics. It is fundamental, in this context, to be acquainted with the notion of heuristic: an experience-based or intuition-guided problem-solving technique, which is generally fast at the price of suboptimality, *i.e.*, there is generally no theorem to fully guarantee the reliability of the results they provide. Heuristics are employed either when just no optimal approach is available at all or when, despite availability of an optimal method, the heuristic approach is way faster and yields results with an acceptable accuracy given the inherent gain in execution time. The reason for the existence of multiple solutions of the same protein structural comparison problem is that disparate inspirations lead to diverse approaches, very often based on very different lines of reasoning. It turns out that the existence of multiple heuristics for protein structure comparison is actually vital and the reason for this is that it is very difficult to establish quantitatively the distance from optimality of suboptimal structural comparison results. Therefore, only when such methods yield results which are in accordance with one another, despite the fact that the nature of the calculations can be profoundly different, can the predictions be deemed accurate.

3. – Applying the Hough Transform to protein comparison

The purpose of the present essay is to illustrate a recently developed heuristic approach for assessing protein structure similarity and for retrieving structural motifs, which is inspired by a computational method imported from the field of computer vision: the Hough Transform [5]. In this method, protein similarity is determined by performing a comparison between pairs of protein structures and calculating a comparison score. Motif retrieval is allowed for by letting one of the proteins in the comparison be the structural motif which is to be searched for. The comparison score is calculated in a vote space, which generally corresponds to the coordinate space in which one of the proteins is described, by appropriately aligning couples of secondary structures of the two proteins (*e.g.*, alpha-helices and beta-strands), one from each of the latter, and voting selected reference points. Either the vote of the mostly voted point in the voting space or other functions of the votes in the voting space can be used as the final comparison score. The method can be easily tuned so that voting is performed only between secondary structures of similar type, *e.g.*, alpha-helices with alpha-helices and beta-strands with beta-strands. Higher information content in the voting procedure always results in a cleaner voting space and in a better signal to noise ratio. The principle of the method is

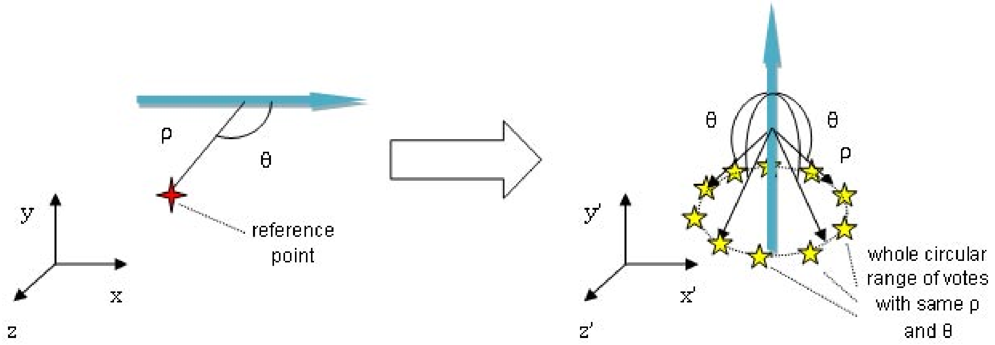


Fig. 1. – The principle of applying the Hough Transform to protein structure comparison. Left: model protein; right: object protein (voting space).

illustrated in fig. 1. On the left is the xyz space of the so-called "model protein", while on the right is the $x'y'z'$ space of the so-called "object protein". Voting is performed in the latter space. In order to do so, for every secondary structure of the model protein (illustrated as a bold arrow on the left of fig. 1), the characteristic parameters rho and theta must be computed; these correspond to the distance of the secondary structure to a well-defined reference point, such as the geometric center of the protein, and the angle that the direction of the secondary structure forms with the segment joining the latter with the reference point. The parameters rho and theta allow to vote reference points in the voting space. Knowledge of only two parameters in a 3D space results in incomplete definition of the position of the point to vote, making voting of an entire circumference indispensable. This circumference is the rim of a cone centered in the object protein's secondary structure, as fig. 1 illustrates. If the secondary structures of the object proteins have a similar spatial arrangement as the ones of the model protein from which the rho and theta parameters are taken from, then, after many voting steps, *i.e.*, when each of the rho and theta values from all of the secondary structures of the model protein has been used to vote circumferences around every secondary structure of the object protein, as many circumferences as the number of secondary structures in the model protein will all intersect, in the voting space, in one highly voted point, which can be used as a score for the comparison. This will not happen whether the two proteins have different structures, resulting in a low score. Figure 2 illustrates a sample voting space, in which successful comparison results in the formation of vote peaks due to the voting circumferences intersecting with one another.

Fundamental for the implementation of the method is discretization. Both the geometric space where voting is performed and the voting circumference are discretized, the former in "voxels" (cubic volume elements), the number of which is called spacemesh, the latter in an integer number of steps (parameter called phimesh). The entity of the discretization deeply influences the quality of the results: a high-detail mesh produces more reliable scores, although at the price of longer execution times, making a compromise between parameter settings and acceptable execution times necessary. Overwhelming memory usage is avoided by maintaining information of only the voxels that contain a nonzero vote. The most important aspect of the implementation regards the way of dealing with the voting space after it has been filled. Since structural similarity does not

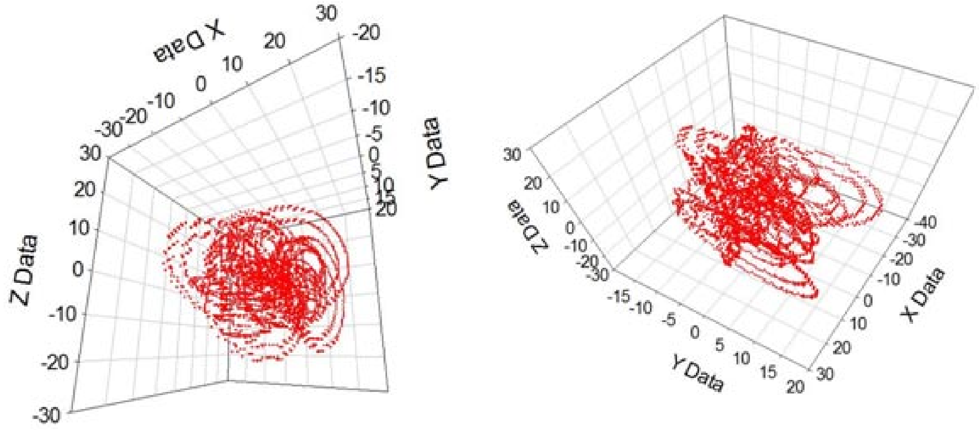
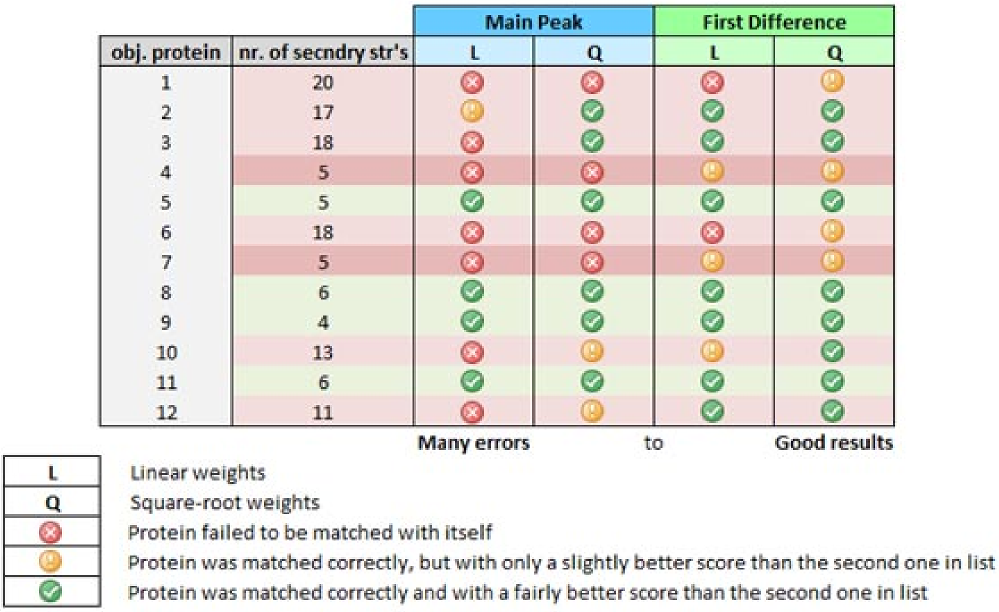


Fig. 2. – Example of a filled voting space, in which circumferences from different secondary structures interfere to create voting peaks (two different perspectives).

mean structural identity, the circumferences in the voting space will not actually overlap perfectly, although they will come to close proximity around one point. This results in the votes not forming the actual peaks that are wanted in order to compute a score. For this reason, smoothing of the vote space by accumulation of all of the votes sufficiently near to every point in the vote space is needed. Performing a smoothing of the vote space



		Main Peak		First Difference	
obj. protein	nr. of secndry str's	L	Q	L	Q
1	20	✗	✗	✗	!
2	17	!	✓	✓	✓
3	18	✗	✓	✓	✓
4	5	✗	✗	!	!
5	5	✓	✓	✓	✓
6	18	✗	✗	✗	!
7	5	✗	✗	!	!
8	6	✓	✓	✓	✓
9	4	✓	✓	✓	✓
10	13	✗	!	!	✓
11	6	✓	✓	✓	✓
12	11	✗	!	✓	✓

Many errors to Good results

L	Linear weights
Q	Square-root weights
✗	Protein failed to be matched with itself
!	Protein was matched correctly, but with only a slightly better score than the second one in list
✓	Protein was matched correctly and with a fairly better score than the second one in list

Fig. 3. – Table with comparative results between different ranking criteria and smoothing-adjustment policies.

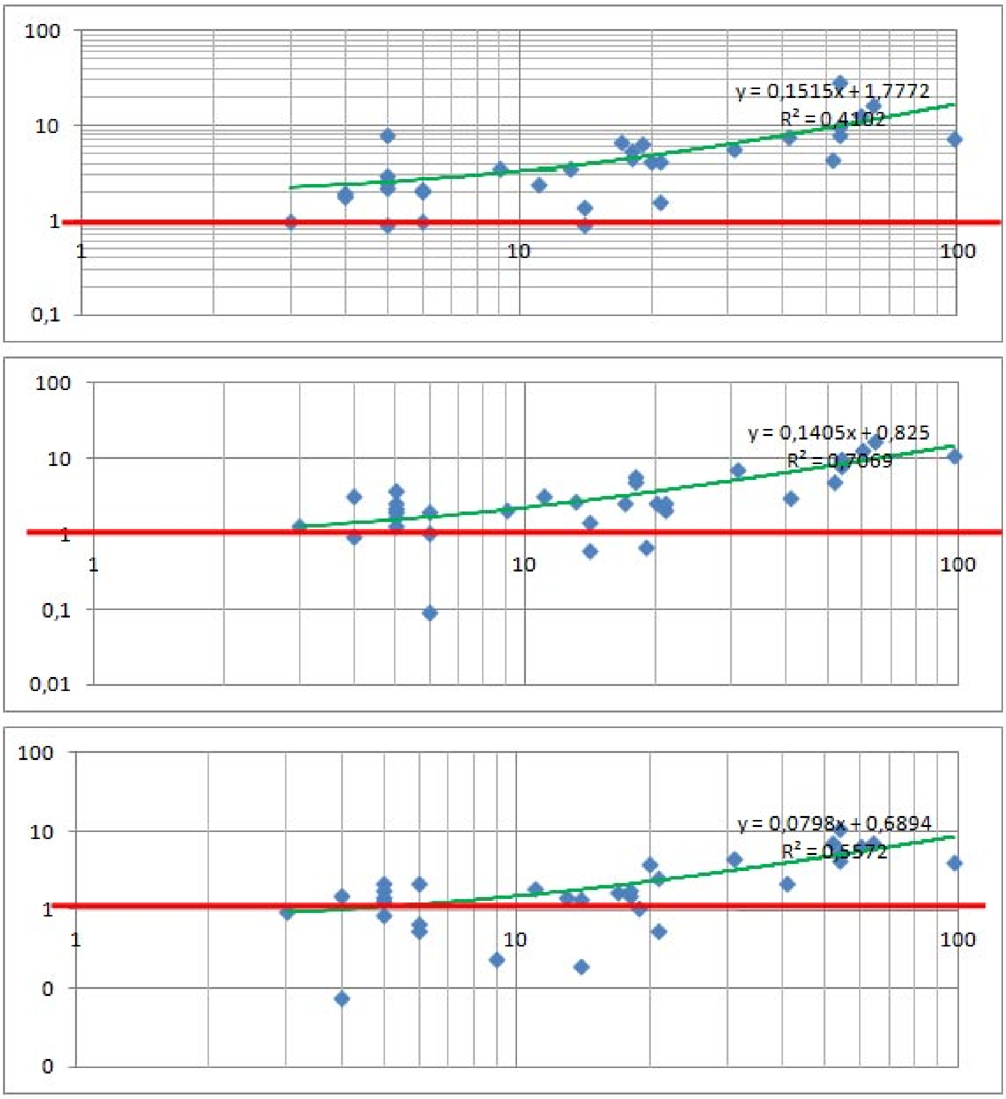


Fig. 4. – Graph of the difference score ratios as a function of object protein size; phimesh = 30, 10, 5 (from top to bottom), spacemesh = $1, 0$.

by merely scrolling the vote space and for every point summing every vote that happens to be sufficiently close to it (thereby scanning the vote space again for every point in it) is computationally costly, to the point that it easily becomes too heavy to be executed in reasonable times for ordinary purposes. The algorithmic complexity is in this case of $O(N2)$, where N is the number of votes in the vote space, which is in turn proportional to the number of secondary structures in the model protein and in the object protein and to the number of steps in which the voting circumference is divided. The problems associated with the high computational requirements of the smoothing algorithm are solved if a particular data structure is introduced in the implementation, *i.e.*, the Range Tree.

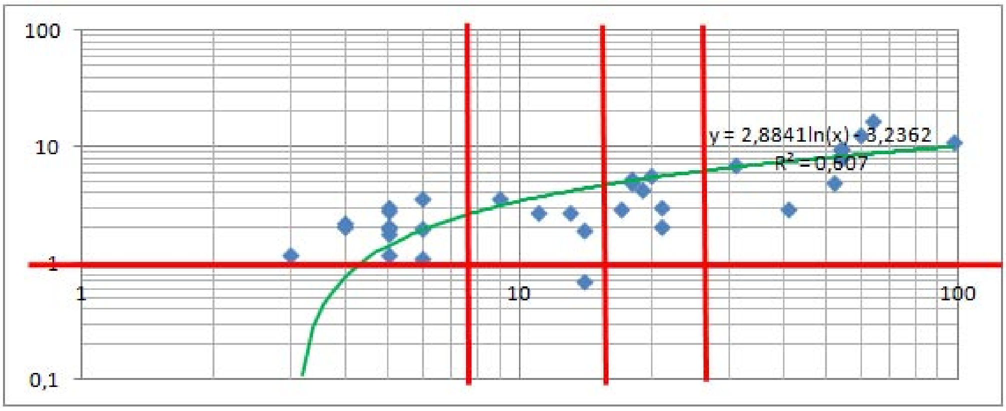


Fig. 5. – Graph of the difference score ratios as a function of object protein size; phimesh: variable (different values separated by bold lines), spacemesh = $1, 0$.

The use of Range Trees in the smoothing step allows the computational complexity to fall down to $O(N \log 3N)$. Balancing of the Range Trees guarantees that the computational complexity stick to the logarithmic order above, without worst-case scenarios. A study (fig. 3) was performed on a 12-to-12 comparison job, which tries to detect proteins in ensembles containing themselves, by ranking the model proteins according to the main peak (the highest vote in the related vote space), shown in the central column. Also in fig. 3 are results with the "first differences", in the rightmost column. These are values that the program computes automatically after each $1 : N$ comparison, and correspond to the differences between the first two peaks in the vote space of each model protein. It is evident that the differences constitute a better way than the raw peak values to indicate whether successful comparison has been achieved: indeed, differences indicate the formation of a dramatically high peak with respect to the "rest" of the peaks in the vote space. Instead, high values may be just the indication of high vote density and not necessarily a peak resulting from a positive GHT result on the queried protein.

A detailed study has been performed in order to understand how the parameters built in the algorithm influence the goodness of the comparisons. As it turned out, different spatial resolutions imply profoundly different results. Figure 4 shows that higher detail results in more reliable comparisons, at the price of higher execution times.

As fig. 5 shows, adaptive modes can also be employed in order to attain a compromise between quality of the results and execution times. Proteins with a higher number of secondary structures are compared with a low detail level, whereas smaller ones need a high resolution, being the ones for which erroneous comparisons are the most likely.

The execution times vary strongly depending on the size of the input, *i.e.*, how many proteins to compare and how many secondary structures they contain, and the parameters settings. A typical execution time for a one-to-one comparison is from a fraction of second to a few seconds on a standard laptop. Noteworthy is the change in execution times that the use of Range Trees brings about: a 33-to-33 proteins comparison took 8 hours to complete with the standard algorithm (without Range Trees), while only 11 minutes with the optimized algorithm (with Range Trees). The algorithm is embarrassingly parallel. This means that, since it requires very little communication between independent voting steps, it is easily split into parallel tasks, resulting in faster execution, which is fundamental for operation in the context of database annotation. All in all, as it has been pointed out in the introductory paragraphs that results of structural alignment from different methods which are in reciprocal agreement help to confirm positive predictions, the Hough Transform method, which has proven successful, is therefore a fundamental component of the suite of protein structural comparison algorithms available to date. The method has its own peculiar advantages, too, which are inherited directly by the algorithm it is based upon, the Hough Transform. Among them, efficiency, rotation and translation insensitiveness, parallelizability, noise and occlusion insensitiveness and relative tolerance to approximate descriptions. The method is also highly parameterized, so that it can be adapted to specific cases in order to obtain the best results. The main downside is its suboptimality, as is the case with any heuristic. Other specific disadvantages of the Hough Transform, such as high time and memory requirements, and of Range Trees, such as the inherent difficulties in treating complex data structures, have been successfully dealt with and solved.

REFERENCES

[1] Taylor W. R. and Orengo C. A., *J. Mol. Biol.*, **208** (1989) 22.

[2] Holm L. and Sander C., *J. Mol. Biol.*, **233** (1993) 123.

[3] Nussinov R. and Wolfson H., *Proc. Natl. Acad. Sci. U.S.A*, **88** (1991) 10495.

[4] Comin M., Guerra C. and Zanotti G., *J. Comput. Biol.*, **11** (2004) 1061.

[5] Mattia E., *Protein structure comparison and motif retrieval by the generalized Hough transform*, Master's thesis, IUSS, Pavia, Italy (2010).